CENTER FOR PARALLEL OPTIMIZATION

MINIMUM-PERIMETER DOMAIN DECOMPOSITION

by

Jonathan Yackel and Robert R. Meyer

Computer Sciences Technical Report #1078

February 1992

Minimum-Perimeter Domain Decomposition *

Jonathan Yackel† Robert R. Meyer†

Abstract

For certain classes of problems defined over two-dimensional regions with grid structure, minimum-perimeter domain decomposition provides tools for partitioning the problem tasks among processors so as to minimize interprocessor communication. Minimizing interprocessor communication is shown to be equivalent to tiling the domain so as to minimize total tile perimeter, where each tile corresponds to the tasks assigned to some processor. The concepts of "slice-convexity" and "semi-perimeter" are introduced to characterize minimum-perimeter tiles. A tight lower bound on the perimeter of a tile as a function of its area is developed. We then show how to generate all possible minimum-perimeter tiles. Certain classes of domains are shown to be optimally tilable.

1 Introduction

Many computations performed on parallel processors involve a collection of tasks which are related by a rectangular grid structure (*i.e.*, as in figure 1, each task has at most four "neighbor" tasks). Examples include the problem of determining the characteristics of fluid flow [5], solving obstacle problems using parallel successive overrelaxation [1], and edge detection in computer vision [6]. We assume initially that all grid cells are squares of uniform size as in figure 1, and that there is a task associated with each cell that uses only its own data and values from neighboring cells that share an edge. For cells on the boundary of the given region, boundary conditions may be used in the computations. If the grid cells are assigned to the processors (that is, the computation for each cell is done by a particular processor), then sharing data with neighboring cells may involve communicating with other processors.

The term "tile" will refer to a connected group of cells assigned to the same processor. We say a set of cells is connected if for every pair of cells c_i, c_j

*This research was partially supported by the Air Force Office of Scientific Research under grant 89-0410, and by the National Science Foundation under grants CCR-8907671 and CCR-8907671.

†Center for Parallel Optimization, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin 53706.

1	1	1					8	8	8
1	1	1					8	8	8
2	2	2					7	7	7
2	2	2	4	4	5	5	7	7	7
3	3	3	4	4	5	5	6	6	6
3	3	3	4	4	5	5	6	6	6

Figure 1: Assigning the grid cells of a domain to processors

there is a *path* of cells in the set from c_i to c_j such that adjacent cells on the path share an edge. (We will show that, in order to achieve the lower bounds on perimeter derived in §3, the cells assigned to each processor must be connected.) To measure interprocessor communication, we measure the length of the tile borders because only across the tile borders may data pass between different processors. In figure 1 we have placed processor identification numbers in the cells to indicate the assignment of cells to processors. For the case depicted in the figure there are eight processors, each assigned six cells for load balancing. Each processor's tile has a perimeter of ten, so the total length of the tile borders is 80 (the results in section 3 show this is the minimum possible total border length for any load-balanced assignment).

This paper thus investigates ways of assigning grid cells to processors so that the total tile perimeter is minimized while the workload is balanced by assigning an appropriate number of cells to each processor.

1.1 Overview

In §2 we present a mathematical statement of the problem. In §3, we develop a lower bound on the optimal value. §4 develops optimal tiles of cells for individual processors, and §5 provides combinations of these tiles producing optimal assignments that attain the lower bound on total perimeter. In §7 we investigate a database application in which domain boundaries are treated differently because a different style of communication is assumed. Our conclusions and future research directions are contained in §8.

2 Problem Statement

Suppose that we wish to allocate the cells of a domain among N processors. Let $\mathcal{A}$ denote the number of cells (area) of the domain. Given a processor p, let $\mathcal{A}_p$ denote the number of cells (or area) assigned to p. ($\mathcal{A}_p$ may also be thought of as the workload assigned to processor p since there is an equal amount of computation associated with each cell.) Load balancing is achieved by constraints specifying a value for $\mathcal{A}_p$ for each processor. (In typical applications, the specified processor loads are equal or differ by at most 1. It is assumed that $\sum_p \mathcal{A}_p = \mathcal{A}$.) We use $\mathcal{P}(T)$ to denote the perimeter of a configuration T of cells. The notation $\mathcal{P}(T_p)$ is used to denote the perimeter of the tile(s) held by processor p. (The cells held by processor p are not necessarily connected, and therefore may comprise several tiles.) The *objective function* $\mathcal{C}$ that we wish to minimize measures interprocessor communication and is defined as follows: $\mathcal{C} := \sum_p \mathcal{P}(T_p)$. This definition is motivated by the assumption that total interprocessor communication may be expressed as the sum of the communication associated with the domain boundary and the communication associated with "interior" borders between tiles (the total length of which is determined by the manner in which cells are assigned to processors). With respect to communication corresponding to the domain boundary, there are at least two possible simplifying assumptions that mesh with the models to be detailed below. One could assume that computing the values of boundary cells is done locally by the processors (*e.g.,* using boundary conditions) with no communication necessary. Alternatively one could assume some fixed amount of communication proportional to boundary edges for each boundary cell. In either case the amount of communication corresponding to the domain boundary is a constant. Thus, the total communication is given by $k_1\mathcal{B}+k_2\sigma$, where $\mathcal{B}$ is the total length of the domain boundary, σ is the total length of the border between tiles (note that each piece of the "interior" border is counted twice, once for each tile), and k_1 and k_2 are scale factors relating boundary and border lengths to communication. Since $\mathcal{B}$ is constant, minimizing this expression is equivalent to minimizing $k_2\mathcal{B}+k_2\sigma$, which in turn is equivalent to minimizing $\mathcal{B}+\sigma$, the total tile perimeter, which is given by $\sum_p \mathcal{P}(T_p)$. The problem, formally stated below, is to minimize total tile perimeter subject to load balancing constraints:

Given: N processors, a domain comprised of grid cells, and a load $\mathcal{A}_p$ for each processor.

Find an assignment that

minimizes	$\mathcal{C}$
s.t.	*every cell is assigned to a processor,*
	and processor p is assigned $\mathcal{A}_p$ cells $(p = 1, 2, \ldots, N)$.

It is easily seen that the number of assignments satisfying the balancing constraint is

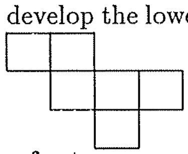
$$\binom{\mathcal{A}}{\mathcal{A}_1}\binom{\mathcal{A}-\mathcal{A}_1}{\mathcal{A}_2}\binom{\mathcal{A}-\mathcal{A}_1-\mathcal{A}_2}{\mathcal{A}_3}\cdots\binom{\mathcal{A}-\mathcal{A}_1-\mathcal{A}_2-\cdots-\mathcal{A}_{N-1}}{\mathcal{A}_N} = \frac{\mathcal{A}!}{\prod_{p=1}^{N}(\mathcal{A}_p!)}.$$

Complete enumeration of these assignments is not feasible even for relatively small problems. For example, given a domain consisting of 25 cells, 5 processors, and a load of 5 for each processor, there are more than 623×10^{12} possible assignments.

The processor assignment problem is related to the graph partition problem which is NP-complete (see Garey and Johnson, page 209 [2]). Our problem is not necessarily NP-complete, however, since the corresponding graph partitioning problem instances are all on graphs with grid structures (each vertex has two, three, or four neighbors).

3 Lower Bounds

In this section we will develop a lower bound on the measure $\mathcal{C}$ and discuss conditions under which this lower bound is attained. To do this we introduce the concept of a configuration's "semi-perimeter", denoted by $\mathcal{S}(C)$, defined as the width plus height of the smallest rectangle enclosing the configuration C. (As discussed below, this is also half the perimeter of the tiles that are used to develop the lower bounds.) For example, the semi-perimeter of the configuration



is $4 + 3 = 7$ and its perimeter is 14. The term "slice" is used to refer to a row or column of either the *domain* or of a *configuration* depending on the context. $\mathcal{S}(C)$ is thus equivalently the number of slices intersecting C. A tight lower bound on $\mathcal{S}(C)$ as a function of the number of cells in C will be developed, yielding a tight lower bound on $\mathcal{P}(C)$, hence a lower bound on $\mathcal{C}$.

3.1 Relation of Semi-perimeter and Perimeter

We introduce the notion of "slice-convexity" of a configuration.

Definition 1 *A configuration is* slice-convex *if for any two cells c_1, c_2 of the configuration in the same slice, the smallest rectangle containing c_1 and c_2 lies entirely in the configuration.*

Lemma 2 *For any configuration C, $\mathcal{P}(C) \geq 2\mathcal{S}(C)$. Furthermore, $\mathcal{P}(C) = 2\mathcal{S}(C)$ if and only if C is a slice-convex configuration.*

Proof: There are at least two edges forming part of the configuration border in each slice of C. Therefore each slice of C contributes at least 2 to the perimeter of C, but exactly 1 to $\mathcal{S}(C)$. Since $\mathcal{S}(C)$ is the number of slices of C, $\mathcal{P}(C) \geq 2\mathcal{S}(C)$. For a slice-convex configuration, each slice of the configuration contains exactly 2 configuration borders in the dimension corresponding to the slice, so that for a slice-convex configuration $\mathcal{P}(C) = 2\mathcal{S}(C)$. For a configuration that is not slice-convex, there is a slice with more than 2 configuration borders, therefore $\mathcal{P}(C) > 2\mathcal{S}(C)$. ∎

When considering minimum-perimeter configurations in the following sections, lemma 2 allows us to restrict our attention to those which are slice-convex.

3.2 Lower Bound on Semi-perimeter

In order to develop this lower bound, we first consider how much area can be enclosed by a given perimeter. Let $\mathcal{A}^*(\mathcal{S})$ be the function mapping $\mathcal{S}$ to the maximum area achievable with semi-perimeter $\mathcal{S}$.

Theorem 3 *Given a semi-perimeter $\mathcal{S}$, the maximum area tile with semi-perimeter $\mathcal{S}$ is an $\frac{\mathcal{S}}{2} \times \frac{\mathcal{S}}{2}$ square if $\mathcal{S}$ is even, and is an $\left(\frac{\mathcal{S}-1}{2}\right) \times \left(\frac{\mathcal{S}+1}{2}\right)$ rectangle if $\mathcal{S}$ is odd,* i.e.,

$$\mathcal{A}^*(\mathcal{S}) = \begin{cases} \left(\frac{\mathcal{S}}{2}\right)^2 & \text{if } \mathcal{S} \text{ is even} \\ \left(\frac{\mathcal{S}-1}{2}\right)\left(\frac{\mathcal{S}+1}{2}\right) & \text{if } \mathcal{S} \text{ is odd} \end{cases}.$$

Proof: For a configuration C let $\mathcal{S}_x(C)$ and $\mathcal{S}_y(C)$ denote the number of columns and number of rows in the configuration respectively. Given any configuration C with semi-perimeter $\mathcal{S}$ and area $\mathcal{A}$, there is a rectangular configuration C' with dimensions $\mathcal{S}_x(C) \times \mathcal{S}_y(C)$, and area $\mathcal{S}_x(C)\mathcal{S}_y(C) \geq \mathcal{A}$. Therefore we need only consider rectangles as candidates for maximum area configurations. To find the rectangle of maximum area with semi-perimeter $\mathcal{S}$, we maximize $\mathcal{S}_x\mathcal{S}_y$ subject to $\mathcal{S}_x + \mathcal{S}_y = \mathcal{S}$. Of all pairs of integers with a certain sum, the pair with the greatest product is the one with the numbers closest together. If $\mathcal{S}$ is even, this is achieved by setting $\mathcal{S}_x = \mathcal{S}_y = \frac{\mathcal{S}}{2}$, and if $\mathcal{S}$ is odd, it is achieved when $\mathcal{S}_x$ and $\mathcal{S}_y$ differ by 1. ∎

By "inverting" the function $\mathcal{A}^*(\mathcal{S})$, we obtain a function $\mathcal{S}^*(\mathcal{A})$ which is defined as *the function mapping area $\mathcal{A}$ to the minimum semi-perimeter of all configurations of $\mathcal{A}$ cells.* We also define $\mathcal{P}^*(\mathcal{A}) := 2\mathcal{S}^*(\mathcal{A})$ as the function mapping an area $\mathcal{A}$ to the minimum perimeter of all configurations of $\mathcal{A}$ cells. (We show below that $\mathcal{S}^*(\mathcal{A})$ is attained by a slice-convex tile, so that by slice-convexity (lemma 2) its perimeter is $2\mathcal{S}^*(\mathcal{A})$.)

Theorem 4

$$\mathcal{S}^*(\mathcal{A}) = i\left\lceil \mathcal{A}^{1/2} \right\rceil + (2-i)\left\lfloor \mathcal{A}^{1/2} \right\rfloor$$

where i is the smallest positive integer such that

$$\left\lceil \mathcal{A}^{1/2} \right\rceil^i \left\lfloor \mathcal{A}^{1/2} \right\rfloor^{2-i} \geq \mathcal{A}.$$

Proof: We may bound the semi-perimeter of any configuration of $\mathcal{A}$ cells from below by finding the smallest semi-perimeter $\mathcal{S}$ that satisfies $\mathcal{A}^*(\mathcal{S}) \geq \mathcal{A}$, since this implies $\mathcal{A} > \mathcal{A}^*(\mathcal{S}-1)$, which means that a semi-perimeter of $\mathcal{S}-1$ is not compatible with an area of $\mathcal{A}$.

Consider the following sequence of rectangles.

$$\begin{array}{ll} Q_0: & 0 \times 0 \\ Q_1: & 1 \times 0 \\ Q_2: & 1 \times 1 \\ Q_3: & 2 \times 1 \\ Q_4: & 2 \times 2 \\ Q_5: & 3 \times 2 \\ Q_6: & 3 \times 3 \\ & \vdots \end{array}$$

We call these rectangles "quasi-squares" since the dimensions of each rectangle differ by at most 1. Note that the areas of the quasi-squares in the sequence are strictly increasing after the second, the area of the ith quasi-square Q_i for $i \geq 2$ is $\mathcal{A}^*(i)$, and the semi-perimeter for the quasi-squares increases by 1 at each step. The areas of these quasi-squares are the points at which the lower bound on the semi-perimeter increases by 1.

For an arbitrary $\mathcal{A}$, there is a unique smallest quasi-square Q_j whose area is at least $\mathcal{A}$. Since the area of Q_j is at least $\mathcal{A}$, by selecting $\mathcal{A}$ cells from Q_j a semi-perimeter of at most $\mathcal{S}(Q_j)$ is achievable for $\mathcal{A}$. Since the area of Q_{j-1} is smaller than $\mathcal{A}$, a semi-perimeter of $\mathcal{S}(Q_{j-1}) = \mathcal{S}(Q_j) - 1$ is not achievable for $\mathcal{A}$. Therefore the smallest semi-perimeter achievable for any configuration of $\mathcal{A}$ cells is $\mathcal{S}(Q_j)$. It is easy to see that each dimension of Q_j is either $\lfloor \mathcal{A}^{1/2} \rfloor$ or $\lceil \mathcal{A}^{1/2} \rceil$ and that $\mathcal{S}(Q_j)$ is exactly the semi-perimeter bound in the statement of the theorem. ∎

The above argument implies a construction technique for "perimeter-optimal" configurations, *i.e.*, configurations with minimum perimeter. An optimal configuration for any $\mathcal{A}$ can be constructed by arranging $\mathcal{A}$ cells into a *partial square* as follows. Start with a complete square with sides of length $\lfloor \mathcal{A}^{1/2} \rfloor$. Add cells to fill in new 1-dimensional faces (completing a face before starting on a new one) until the total number of cells is $\mathcal{A}$. The resulting partial square will have sides of length $\lfloor \mathcal{A}^{1/2} \rfloor$ and $\lceil \mathcal{A}^{1/2} \rceil$, and will measure $\lceil \mathcal{A}^{1/2} \rceil$ in as few dimensions as possible. By theorem 4, it will have minimum semi-perimeter. If the partial squares are constructed to be slice-convex then by lemma 2 they also have minimum perimeter. This construction technique is a special case of a technique to be described in §4.2 for constructing all minimum-perimeter configurations of a given area.

In figure 2, we show some perimeter-optimal partial squares constructed in this manner with areas ranging from one to sixteen.

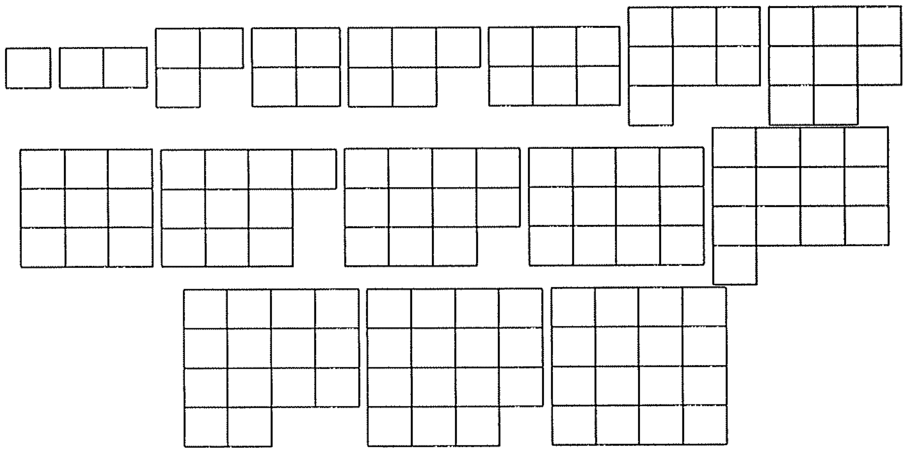


Figure 2: Partial squares with minimum perimeter

In §3 of Ghandeharizadeh *et al* [3], we derived an alternate bound on $\mathcal{S}$:

$$\mathcal{S} \geq \left\lceil 2\,\mathcal{A}^{1/2} \right\rceil .$$

This bound is equivalent to $\mathcal{S}^*(\mathcal{A})$ (see Yackel and Meyer [7]).

Table 1 constructed via theorem 4 contains minimum perimeter values for areas up to 56.

3.3 Lower Bound on $\mathcal{C}$

Corollary 5 $\displaystyle\sum_p \mathcal{P}^*(\mathcal{A}_p) \leq \mathcal{C}$.

Proof: Use theorem 4 and the fact that $\mathcal{C} = \sum_p \mathcal{P}(T_p)$. ∎

Clearly, if the configuration for each processor has minimum perimeter (*i.e.*, $\mathcal{P}(\mathcal{A}_p) = \mathcal{P}^*(\mathcal{A}_p)$ for all p), then the corresponding set of cell assignments achieves the lower bound on the communication measure $\mathcal{C}$, and is therefore an optimal assignment.

In §5 we give classes of domains for which such assignments are possible.

In the following lemma we present a perimeter "optimality test" for any configuration.

Lemma 6 *A configuration of $\mathcal{A}$ cells with semi-perimeter $\mathcal{S} > 2$ has minimum perimeter if and only if it is a slice-convex tile satisfying*

$$\mathcal{A}^*(\mathcal{S} - 1) < \mathcal{A} . \qquad (1)$$

min perimeter ($\mathcal{P}^*(\mathcal{A})$)	area ($\mathcal{A}$)
4	1
6	2
8	3 – 4
10	5 – 6
12	7 – 9
14	10 – 12
16	13 – 16
18	17 – 20
20	21 – 25
22	26 – 30
24	31 – 36
26	37 – 42
28	43 – 49
30	50 – 56

Table 1: Minimum perimeters

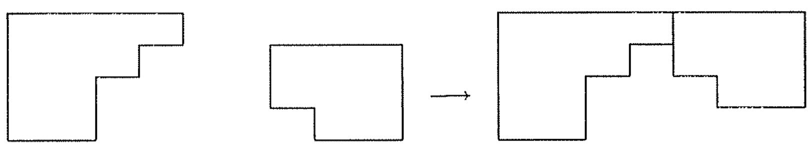


Figure 3: translating sub-tiles to decrease perimeter

Proof: From lemma 2 it follows that only slice-convex configurations may have minimum perimeter. To see that only tiles (*i.e.*, connected configurations) may have minimum perimeter, consider a configuration containing two disconnected sub-tiles denoted by T_1 and T_2. By translating sub-tile T_1 it is always possible to connect T_1 and T_2 (see figure 3), thereby decreasing the perimeter of the configuration by at least 2. If (1) holds, then $\mathcal{A}$ is greater than the largest area for which a smaller semi-perimeter is achievable, so the configuration has minimum semi-perimeter. This, along with slice-convexity imply the configuration has minimum perimeter. ∎

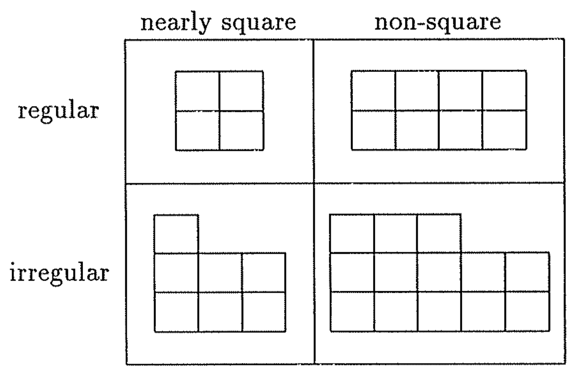


Figure 4: Categories of optimal configurations

4 Additional Configurations with Minimum Perimeter

In this section we discuss some additional characteristics of configurations which have minimum perimeter for their area. The previous section establishes that configurations that are square or nearly square have minimum perimeter. In this section we will see that configurations of other shapes may also have minimum perimeter. We may classify configuration shapes according to two independent characteristics. Configurations are either nearly square (dimensions differing by at most 1) or non-square. In addition, configurations are either regular (complete rectangles) or irregular. Figure 4 depicts examples of minimum-perimeter configurations in each of the four categories induced by these characteristics.

Non-square regular configurations (rectangles) with minimum perimeter are discussed in §4.1. Irregular minimum-perimeter configurations and a technique for constructing all optimal configurations of a given area are taken up in §4.2.

4.1 Optimal Rectangles

Using the results from the previous section, we can characterize the rectangular blocks that have minimum perimeter.

Theorem 7 *An $x \times (x+k)$ or an $(x+k) \times x$ rectangular block is perimeter-optimal if and only if*

$$k \text{ is even and } 1 + \tfrac{k}{2}(\tfrac{k}{2} - 1) \leq x$$
$$\text{or}$$
$$k \text{ is odd and } 1 + \left(\tfrac{k-1}{2}\right)^2 \leq x.$$

Conversely, an $x \times (x+k)$ or an $(x+k) \times x$ rectangle is perimeter-optimal if and only if the rectangularity increment k is at most

$$\max\left\{2\ round(x^{1/2}), 2\left\lfloor x^{1/2}-1\right\rfloor+1\right\}$$

where $round(x)$ rounds x to the nearest integer.

Proof: To prove the first part of the theorem, we simply apply the optimality test. By lemma 6, an $x \times (x+k)$ block is optimal if and only if

$$\left\lceil \frac{2x+k-1}{2} \right\rceil^{r} \left\lfloor \frac{2x+k-1}{2} \right\rfloor^{2-r} < x^2+kx \tag{2}$$

where $r \equiv 2x+k-1 \pmod 2$.

If k is even, (2) reduces to

$$\begin{array}{rccc} & \left\lceil\frac{2x+k-1}{2}\right\rceil\left\lfloor\frac{2x+k-1}{2}\right\rfloor & < & x^2+kx \\ \Longleftrightarrow & (x+\frac{k}{2})(x+\frac{k}{2}-1) & < & x^2+kx \\ \Longleftrightarrow & \frac{k}{2}(\frac{k}{2}-1) & < & x. \end{array}$$

The integrality of both sides of the inequality allow us to derive the desired result.

If k is odd, (2) reduces to

$$\begin{array}{rccc} & \left\lfloor\frac{2x+k-1}{2}\right\rfloor^2 & < & x^2+kx \\ \Longleftrightarrow & \left(x+\frac{k-1}{2}\right)^2 & < & x^2+kx \\ \Longleftrightarrow & \left(\frac{k-1}{2}\right)^2 & < & x. \end{array}$$

To prove the second part of the theorem, we show that $2\left\lfloor(x-1)^{1/2}\right\rfloor+1$ and $2\ \mathrm{round}\left(x^{1/2}\right)$ are the largest odd and even integers respectively satisfying (2).

To prove the result for the odd numbers, we start with the expression for x in terms of odd k.

$$\begin{array}{rccc} & \left(\frac{k-1}{2}\right)^2+1 & \leq & x \\ \Longleftrightarrow & \frac{k-1}{2} & \leq & (x-1)^{1/2}. \end{array}$$

Since the LHS of the last inequality is integer, we may take the floor of the RHS.

$$\begin{array}{rccc} \Longleftrightarrow & \frac{k-1}{2} & \leq & \left\lfloor(x-1)^{1/2}\right\rfloor \\ \Longleftrightarrow & k & \leq & 2\left\lfloor(x-1)^{1/2}\right\rfloor+1. \end{array}$$

Since the RHS of the last inequality is an odd integer, $k = 2\left\lfloor(x-1)^{1/2}\right\rfloor+1$ is the largest odd integer satisfying (2).

To prove the result for the even numbers we write $x^{1/2}$ in the form $x = r+f$ where r is the integer part and $f \in [0,1)$ is the fractional part. If $f < \frac{1}{2}$ then

2 round $\left(x^{1/2}\right) = 2r$. If $f > \frac{1}{2}$ then 2 round $\left(x^{1/2}\right) = 2r + 2$. (For integer x, f is never $\frac{1}{2}$ so round $\left(x^{1/2}\right)$ is uniquely defined for integer x.)

If $f < \frac{1}{2}$ and 2 round $\left(x^{1/2}\right) = 2r$ then $k = 2r$ satisfies (2) because

$$\frac{k}{2}\left(\frac{k}{2} - 1\right) = r(r-1) = r^2 - r < r^2 + 2fr + f^2 = x$$

and $k = 2r + 2$ violates (2) because

$$\frac{k}{2}\left(\frac{k}{2} - 1\right) + 1 = (r+1)r + 1 = r^2 + r + 1 > r^2 + 2fr + f^2 = x.$$

If $f > \frac{1}{2}$ and 2 round $\left(x^{1/2}\right) = 2r + 2$ then $k = 2r + 2$ satisfies (2) because

$$\frac{k}{2}\left(\frac{k}{2} - 1\right) = (r+1)r = r^2 + r < r^2 + 2fr + f^2 = x$$

and $k = 2r + 4$ violates (2) because

$$\frac{k}{2}\left(\frac{k}{2} - 1\right) = (r+2)(r+1) = r^2 + 3r + 2 > r^2 + 2fr + f^2 = x.$$

Therefore $k = 2$ round $\left(x^{1/2}\right)$ is the largest even integer satisfying (2). ∎

Note that the first part of the theorem shows that if a particular rectangle is optimal, then by increasing both dimensions by the same amount, the resulting larger rectangle is also optimal. Theorem 7 is addressed graphically in figure 5, which shows the dimensions of all rectangles with $x \leq 30$ that have minimum perimeter. The integral points on the diagonal line in the figure represent the squares, and the outer boxes represent the most-skewed rectangles with optimal perimeter. All integer points between (and including) the boxed points correspond to rectangles with minimum perimeter. Table 2 lists dimensions of the most skewed optimal rectangles corresponding to the boxed points above the diagonal.

4.2 Optimal Irregular Tiles

In order to characterize irregular optimal configurations we prove a simple but powerful theorem giving another necessary and sufficient condition for the perimeter optimality of *any* configuration.

Theorem 8 *A configuration of $\mathcal{A}$ cells has minimum perimeter if and only if it is slice-convex and its minimum circumscribing rectangle has perimeter $\mathcal{P}^*(\mathcal{A})$.*

Proof: The theorem follows from the fact that a rectangle has the same perimeter as any slice-convex configuration of cells it minimally circumscribes. ∎

Theorem 4.2 suggests a way to find *all* the minimum-perimeter configurations for a given area $\mathcal{A}$. In this discussion we consider two configurations to be

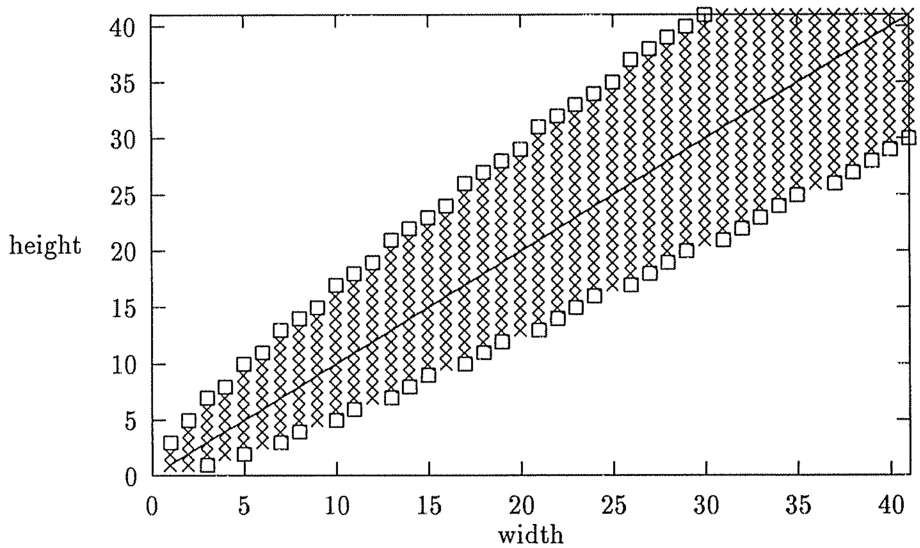
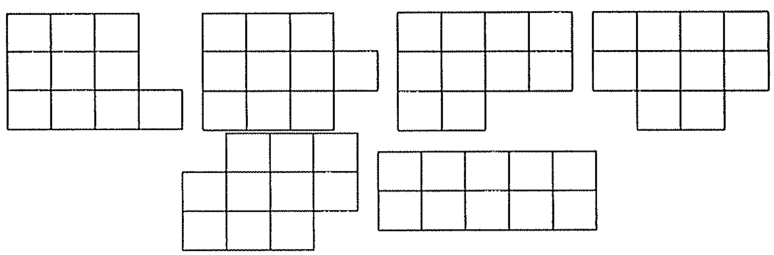


Figure 5: Dimensions of rectangles with optimal perimeter

equivalent if one can be transformed into the other by rotation and/or reflection. Using the theorem, all the possible circumscribing rectangles for perimeter-optimal configurations of area $\mathcal{A}$ can be specified. For each of these rectangles, any slice-convex subset of $\mathcal{A}$ cells forms a minimum-perimeter configuration. We say a cell is a *corner cell* of a configuration if it is not between two cells in any slice. By removing a corner cell from a slice-convex configuration, convexity is maintained. Therefore, starting from a rectangle, a minimum-perimeter configuration can be constructed by iteratively removing corner cells. For example, given an area of 10, the minimum-perimeter configurations are generated as follows. $\mathcal{S}^*(10) = 7$, so the rectangles of semi-perimeter 7 are considered. The possibilities are 1×6, 2×5, and 3×4. A 1×6 rectangle can't enclose 10 cells, so all minimum-perimeter configurations are circumscribed by either 2×5 or 3×4 rectangles. Therefore all minimum-perimeter configurations of area 10 are equivalent to the following configurations:

k	Most-skewed perimeter-optimal rectangles $x \times (x+k)$				
2	1×3				
3	2×5				
4	3×7	4×8			
5	5×10	6×11			
6	7×13	8×14	9×15		
7	10×17	11×18	12×19		
8	13×21	14×22	15×23	16×24	
9	17×26	18×27	19×28	20×29	
10	21×31	22×32	23×33	24×34	25×35
11	26×37	27×38	28×39	29×40	30×41

Table 2: Some most-skewed perimeter-optimal rectangles



The first five configurations above represent all the possibilities for configurations enclosed by 3×4 rectangles: the first two are constructed by removing two corners from the same column, the next two by removing two corners from the same row, and the fifth by removing two corners from different rows and columns. There is only one possible configuration of area 10 contained in a 2×5 rectangle.

By examining the above technique, we are able to identify the cases in which there is a unique minimum-perimeter configuration of given area. Given an area $\mathcal{A}$, if there is a unique rectangle with semi-perimeter $\mathcal{S}^*(\mathcal{A})$ and area $\geq \mathcal{A}$, then all minimum-perimeter configurations of area $\mathcal{A}$ are circumscribed by that rectangle. Furthermore, if the area of that unique enclosing rectangle is $\mathcal{A}$ or $\mathcal{A}+1$ then there is a unique optimal configuration of area $\mathcal{A}$, because all removals of either zero or one corner result in equivalent configurations. We also show that all other cases lead to non-uniqueness.

Lemma 9 *There is a unique minimum-perimeter configuration of area $\mathcal{A}$ if and only if $\mathcal{A}$ can be expressed as $\mathcal{A} = k^2$, $k(k+1)$, or $k(k+1)-1$ for positive integer k.*

Proof: If $\mathcal{A} = k^2$ then $\mathcal{S}^*(\mathcal{A}) = 2k$. The unique enclosing rectangle with semi-perimeter $2k$ and area at least $\mathcal{A}$ has dimensions $k \times k$ (any other rectangle with semi-perimeter $2k$ has area less than k^2). Similarly, if $\mathcal{A} = k(k+1)$, or $k(k+1) - 1$ and $k \geq 2$, then $\mathcal{S}^*(\mathcal{A}) = 2k + 1$, and the unique rectangle with semi-perimeter $2k + 1$ and area at least $\mathcal{A}$ has dimensions $k \times (k+1)$.

To show that these are the only classes of areas which have unique optimal configurations, consider the fact that such an area $\mathcal{A}$ must have only one possible enclosing rectangle with perimeter $\mathcal{S}^*(\mathcal{A})$. Since $\mathcal{S}^*(\mathcal{A})$ corresponds to an enclosing square or quasi-square, it follows that the unique enclosing rectangle must be this square or quasi-square. This means that the area $\mathcal{A}$ is expressible as either $k^2 - j$ or $k(k+1) - j$ for positive integer k and non-negative integer j. For areas expressed as $\mathcal{A} = k^2 - j$ with $j \geq 1$, a $(k+1) \times (k-1)$ rectangle is a second enclosing rectangle with semi-perimeter $2k$ because $(k+1)(k-1) \geq k^2 - j$. For areas expressed as $k(k+1) - j$ with $j > 2$ there can not be a unique optimal configuration because there are at least two ways of removing j corners from a $k \times (k+1)$ rectangle. ∎

The preceding argument proves that squares, quasi-squares, and quasi-squares minus one corner are the unique optimal configurations for their areas and that all other areas have alternate optimal configurations. Similar reasoning can be used to show that for areas of the form $k^2 - 1$ with $k \geq 2$, there are exactly two optimal configurations: a $k \times k$ square with one corner removed and a complete $(k-1) \times (k+1)$ rectangle.

Another interesting fact about the set of optimal configurations of a given area is that there can be at most one rectangular configuration in the set. To prove this we make use of the following lemma.

Lemma 10 *For (unordered) pairs of positive numbers, two distinct pairs with identical pair sums have different pair products,* i.e., *for positive numbers* x_1, y_1, x_2, y_2, *if* $x_1 + y_1 = x_2 + y_2$ *and* $\{x_1, y_1\} \neq \{x_2, y_2\}$ *then* $x_1 y_1 \neq x_2 y_2$.

Proof: Write x_2 as $x_1 + k$. Then $y_2 = y_1 - k$. Now assume $x_1 y_1 = x_2 y_2$. Substituting into this last equation, we get

$$x_1 y_1 = (x_1 + k)(y_1 - k).$$

Simplifying, we get

$$k(y_1 - x_1 - k) = 0,$$

in other words $k = 0$ or $k = y_1 - x_1$. In the first case $x_1 = x_2$ and $y_1 = y_2$, and in the second case $x_1 = y_2$ and $y_1 = x_2$, a contradiction. ∎

The lemma tells us that all rectangles with a given semi-perimeter have different areas and implies that if a rectangular configuration of area $\mathcal{A}$ is optimal then it is the only optimal rectangular configuration with area $\mathcal{A}$ (any other optimal configurations will be irregular).

5 Optimal Tilings

In order to achieve the lower bound for $\mathcal{C}$, each configuration must have perimeter exactly $\mathcal{P}^*(\mathcal{A}_p)$. Thus, we wish to interleave perimeter-optimal configurations for all processors in order to fill the domain exactly. In this section we exhibit classes of domains for which such tilings can be constructed.

5.1 Optimal Tilings with Rectangles

One class of problems that have easily obtainable optimal solutions are instances in which the domain is a $M_1 \times M_2$ rectangular grid that can be tiled with perimeter-optimal rectangles. In particular, if N can be factored as $f_1 f_2$ where f_1 divides M_1, f_2 divides M_2 and $\frac{M_1}{f_1} \times \frac{M_2}{f_2}$ rectangles are perimeter-optimal, then such a tiling is possible when all processors have equal loads. Below we demonstrate an optimal assignment for such an instance: a 6×18 grid with 6 processors, each of which has a load of 18.

1	1	1	1	1	1	2	2	2	2	2	2	3	3	3	3	3	3
1	1	1	1	1	1	2	2	2	2	2	2	3	3	3	3	3	3
1	1	1	1	1	1	2	2	2	2	2	2	3	3	3	3	3	3
4	4	4	4	4	4	5	5	5	5	5	5	6	6	6	6	6	6
4	4	4	4	4	4	5	5	5	5	5	5	6	6	6	6	6	6
4	4	4	4	4	4	5	5	5	5	5	5	6	6	6	6	6	6

However, it is not necessary that all the tiles be oriented in the same way. An alternate optimal assignment for the same problem is shown below.

1	1	1	2	2	2	2	2	2	3	3	3	4	4	4	4	4	4
1	1	1	2	2	2	2	2	2	3	3	3	4	4	4	4	4	4
1	1	1	2	2	2	2	2	2	3	3	3	4	4	4	4	4	4
1	1	1	5	5	5	5	5	5	3	3	3	6	6	6	6	6	6
1	1	1	5	5	5	5	5	5	3	3	3	6	6	6	6	6	6
1	1	1	5	5	5	5	5	5	3	3	3	6	6	6	6	6	6

Figure 1 in §1 is an example of a non-rectangular domain optimally tiled with rectangles.

5.2 Optimal Tiling with Irregular Tiles

Irregular tiles can fit together to tile many grids. The example below shows how irregular optimal tiles of area 10 fit nicely together.

1	1	1	1	4	4	4	4	5	5	6	6	6
1	1	1	1	4	4	4	4	5	5	6	6	6
1	1	2	2	3	3	4	4	5	5	5	6	6
2	2	2	2	3	3	3	3	5	5	5	6	6
2	2	2	2	3	3	3	3	8	8	8	7	7
								8	8	8	7	7
								8	8	7	7	7
								8	8	7	7	7

This example also demonstrates the technique of optimally tiling by decomposing the domain into subdomains which can each be optimally tiled by a proportional subset of the processors. The domain above can be split into four 5×4 rectangles, each of which can be tiled optimally with two processors. Such a divide and conquer approach to tiling is a method of constructing optimal tilings for large domains with complex shapes.

6 Non-uniform Grids

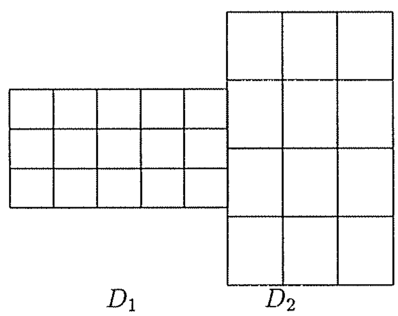


Figure 6: A non-uniform grid

Under certain assumptions, the tiling approaches discussed above can also be used to deal with non-uniform grids which are important in many applications (see *e.g.*, Gropp and Keyes [5]). Considering the case shown in figure 6, suppose that that N processors are available, that the amount of computation per cell (regardless of cell size) is uniform, that total computing loads are to be nearly balanced among processors, and, finally, to ensure relative simplicity and uniformity of the computational procedure implemented on each processor, we impose the constraint that the cells assigned to each processor be of uniform size (i.e., that the tile associated with a processor lie entirely in subdomain D_1 or D_2). In order to first set up load balancing constraints, we would then partition the given number of processors between the two domains so that the number of cells per processor (rather than the area spanned by the processor) was approximately equal. In the second stage of this process, communication would be independently minimized for each subdomain by tiling (using its pre-determined appropriate number of processors) according to the procedures described above for minimum-perimeter tilings. The shared boundaries of the subdomains, across which communication would be required, would be automatically included in the objective function via the boundaries of the individual subdomains. Of course, this two-stage approach is easily generalized to provide a decomposition of arbitrary unions of subdomains into single subdomain problems of the type previously considered.

7 Toroidal Domains

So far, the boundary of the domain has contributed a constant term to the objective function since it has always formed part of the tile borders. In some applications, however, such as the database problem discussed in Ghandeharizadeh *et al* [4], the boundary of the domain is irrelevant to the communication measure. In this database application, we wish to assign grid cells of a rectangular domain to processors in order to minimize the number of *distinct* processors that appear in the slices of the grid. A typical computation in the database system accesses all the data in a particular slice of the grid, and the processors assigned to cells in the slice must participate by communicating with a *coordinating* processor. We assume a communication overhead is associated with initiating and terminating a query on each of the processors associated with a slice. The goal is to minimize overhead while balancing the workload between the processors. If we define ν_s to be the number of distinct processors in slice s of the grid, then our objective is to minimize $\theta_{\text{total}} := \sum_s \nu_s$. In Ghandeharizadeh *et al* [4] we showed that this is equivalent to minimizing the sum of the "D-perimeters" for the processors, where the D-perimeter for a processor is defined to be the number of slices that a processor appears in. In two dimensions, the D-perimeter of a configuration of cells is therefore the semi-perimeter of the configuration in its most "compact" form (a configuration is compact if its slices are adjacent – see figure 7). Because θ_{total} is not affected by a permutation of the rows or columns of the grid, the cells belonging to a processor do not have to be slice-convex or even connected in order to form a configuration with minimum D-perimeter (see again figure 7). However, minimum-perimeter tiles are examples of configurations that have minimum D-perimeter.

We present an optimal solution to the database problem for a certain class of grids. Because of the nature of the problem, the tiles can "wrap around" the top or side of the grid without incurring any extra overhead since they remain in the same slices. We can therefore think of the domain as lying on the

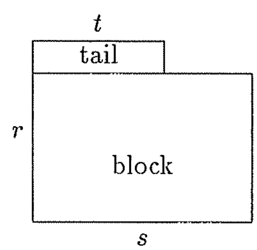
p	p	p						
p	p	p						
p	p	p						

	p			p		p		
	p			p		p		
	p			p		p		

Figure 7: Compact and non-compact forms of a minimum D-perimeter configuration

surface of a torus, *i.e.*, the top row of the grid is adjacent to the bottom row, and the left-most column is adjacent to the right-most. So, using the concept of connectedness in the toroidal sense, the closest analogy to the preceding model is obtained by tiling toroidal domains with slice-convex minimum perimeter tiles. Tilings of this form provide optimal solutions to the above database application (see Yackel and Meyer [7]).

Consider an $N \times N$ toroidal domain, with equal areas assigned to N processors so that $\mathcal{A}_p = N$, $(p = 1, 2, \ldots, N)$. Under these constraints it is always possible to tile the domain with minimum-perimeter tiles. Let $r = \lfloor \sqrt{N} \rfloor$, $s = \max\{k | rk \leq N\}$ and $N = rs + t$, $0 \leq t < s$. Note that $s - r$ is either 0 or 1. Consider the partial square tile of area N (having minimum perimeter), made up of an $r \times s$ block and a "tail" of length t.

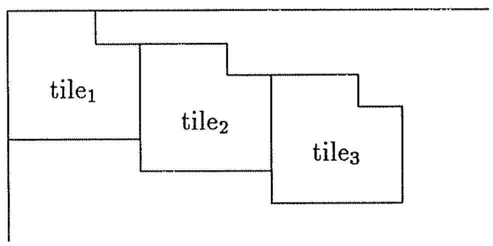


We describe how to tile the grid with these partial squares. If the tail is of length 0, then the grid can be tiled as described at the beginning of section 5.1 by $r \times s$ rectangular tiles. (This is true because $N = rs + 0$.)

If the tail length is not 0, then diagonal tiling (with wrap-around) partitions the grid into tiles as follows. The first tile in the grid's upper left corner is made up of the set of cells:

$$\begin{aligned} \text{tile}_1 = \{ \quad & (1,1), (1,2), \ldots, (1,t), \\ & (2,1), (2,2), \ldots, (2,s), \\ & \vdots \\ & (r+1,1), (r+1,2), \ldots, (r+1,s)\} \ . \end{aligned}$$

The coordinates of the cells in the $(i+1)$st tile are obtained by adding i to all the row coordinates and is to all the column coordinates of the cells in tile_1 modulo N, as shown below.



Theorem 11 *The N tiles $tile_1$, $tile_2$, ..., $tile_N$ cover the grid completely with no overlap.*

Proof: Since each tile contains N cells and the grid has N^2 cells, it suffices to prove that each each grid cell belongs to at least one of tile_1, tile_2,..., tile_N, *i.e.*, for any grid cell (x, y)

$$x \equiv x_1 + i \pmod N$$

and

$$y \equiv y_1 + is \pmod N$$

for some $0 \leq i \leq N-1$, and some $(x_1, y_1) \in \text{tile}_1$. Assume by way of contradiction that there exists some grid cell (x, y) such that for any $(x_1, y_1) \in \text{tile}_1$

$$x \not\equiv x_1 + i \pmod N$$

or

$$y \not\equiv y_1 + is \pmod N$$

for $i = 0, 1, \ldots, N-1$. For each $(x_1, y_1) \in \text{tile}_1$, $x \equiv x_1 + (x - x_1)\ (N)$ so our assumption forces the incongruence

$$y \not\equiv y_1 + (x - x_1)s \pmod N.$$

Therefore the following set of N incongruences (one for each cell in tile_1

moving from right to left, starting in row 1) must hold:

$$
\begin{aligned}
y &\not\equiv t + (x-1)s \pmod N \\
y &\not\equiv t - 1 + (x-1)s \pmod N \\
&\vdots \\
y &\not\equiv 1 + (x-1)s \pmod N \\
\\
y &\not\equiv s + (x-2)s \pmod N \\
y &\not\equiv s - 1 + (x-2)s \pmod N \\
&\vdots \\
y &\not\equiv 1 + (x-2)s \pmod N \\
\\
&\vdots \\
\\
y &\not\equiv s + (x-r-1)s \pmod N \\
y &\not\equiv s - 1 + (x-r-1)s \pmod N \\
&\vdots \\
y &\not\equiv 1 + (x-r-1)s \pmod N.
\end{aligned}
$$

The right hand sides of the above incongruences form a sequence of N consecutive integers.

Thus, we have $y \not\equiv i \pmod N$, $i = 1, 2, \ldots, N$, a contradiction. ∎

Figure 7 illustrates diagonal tiling on a 7×7 grid with 7 processors.

1	6	6	6	7	7	7
1	1	1	2	7	7	7
1	1	1	2	2	2	3
3	3	4	2	2	2	3
3	3	4	4	4	5	3
5	6	4	4	4	5	5
5	6	6	6	7	5	5

Figure 8: Diagonal tiling of a toroidal domain

8 Conclusions and Future Work

We have formalized the problem of partitioning tasks among processors for parallel domain decomposition computations in order to minimize interprocessor communication. A lower bound on the objective function has been developed and we have demonstrated how the bound is attained when the domain can be tiled with minimum-perimeter tiles. We have presented characteristics of minimum-perimeter tiles and systematic techniques for generating all minimum-perimeter tiles with a given area. Finally, we have shown how certain domains can be tiled with minimum perimeter tiles, therefore providing optimal solutions to the communication problem. Continuing work in this area includes developing algorithms to generate optimal or near-optimal solutions for arbitrary domains and numbers of processors. Lower bounds on the objective function taking into account the shape of the domain would allow verification of optimality for solutions to a wider range of problems and sharpen the results in this paper. Extensions of the results to three-dimensional domains and other data partitioning problems is also a goal.

References

[1] R. DeLeone and M. A. Tork-Roth, *Massively parallel solution of quadratic programs via successive overrelaxation*, Computer Sciences Technical Report 1041, University of Wisconsin - Madison, Madison, WI, August 1991.

[2] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman and Company, New York, 1979, pp. 60–62.

[3] S. Ghandeharizadeh, G. L. Schultz, R. R. Meyer, and J. Yackel, *Optimal balanced assignments and a parallel database application*, Computer Sciences Technical Report 986, University of Wisconsin - Madison, Madison, WI, December 1990.

[4] ———, *Optimal processor assignment for parallel database design*, Computer Sciences Technical Report 1022, University of Wisconsin - Madison, Madison, WI, May 1991.

[5] W. D. Gropp and D. E. Keyes, *Domain decomposition methods in computational fluid dynamics*, Tech. Report 91-20, ICASE, February 1991.

[6] R. Schalkoff, *Digital Image Processing and Computer Vision*, John Wiley & Sons, Inc., 1989.

[7] J. Yackel and R. R. Meyer, *Optimal tilings for parallel database design*, Computer Sciences Technical Report 1046, University of Wisconsin - Madison, Madison, WI, September 1991. To appear in Advances in Optimization and Parallel Computing, P. M. Pardalos. ed., North-Holland.