JOURNAL OF PARALLEL AND DISTRIBUTED COMPUTING 6, 649–662 (1989)

Divide and Conquer under Global Constraints: A Solution to the N-Queens Problem

BRUCE ABRAMSON[1]

Department of Computer Science, University of Southern California, Los Angeles, California 90089-0782

AND

MOTI YUNG[2]

Department of Computer Science, Columbia University, New York, New York 10027

Received February 10, 1987

Configuring N mutually nonattacking queens on an N-by-N chessboard is a classical problem that was first posed over a century ago. Over the past few decades, this problem has become important to computer scientists by serving as the standard example of a globally constrained problem which is solvable using backtracking search methods. A related problem, placing the N queens on a toroidal board, has been discussed in detail by Póyla and Chandra. Their work focused on characterizing the solvable cases and finding solutions which arrange the queens in a regular pattern. This paper describes a new *divide-and-conquer* algorithm that solves both problems and investigates the relationship between them. The connection between the solutions of the two problems illustrates an important, but frequently overlooked, method of algorithm design: detailed combinatorial analysis of an overconstrained variation can reveal solutions to the corresponding original problem. The solution is an example of solving a globally constrained problem using the divide-and-conquer technique, rather than the usual backtracking algorithm. The former is much faster in both sequential and parallel environments.

[1] Supported in part by NSF Grant IST-8418879.

[2] Supported in part by NSF Grant MCS-8303139 and an IBM fellowship. Current address: IBM Research Division, T.J. Watson Research Center, Yorktown Heights, NY 10598.

0743-7315/89 $3.00

> The problem of the eight queens is a well known example of the use of trial-and-error methods and of backtracking algorithms. It was investigated by C. F. Gauss in 1850, but he did not completely solve it. This should not surprise anyone. After all, the characteristic property of these problems is that they defy analytic solution. Instead, they require large amounts of exacting labor, patience and accuracy. Such algorithms have therefore gained relevance almost exclusively through the automatic computer, which possesses these properties to a much higher degree than people, even geniuses, do. [23, p. 143]

1. Introduction

The preceding quote typifies the prevalent view of the N-queens problem. First posed in 1848, the problem of placing eight queens on an eight-by-eight chessboard in mutually nonattacking positions was investigated by several 19th-century mathematicians. The more general N-queens problem was solved in the early 1950s by Yaglom and Yaglom [24], who found a pair of patterns which yield exactly one solution for each N (see Appendix). Their patterns, which were not derived algorithmically and offer no insights into the combinatorial nature of the problem, remain relatively unknown. For details of the problem's early history see [1, 9, 16].

Recent work on the N-queens problem has identified three variants: finding one solution, finding a family of solutions, and finding all solutions. The inherent difficulty of all three is generally accepted, and researchers have either used the problem to demonstrate trial-and-error methods like backtracking or modified the problem to one that does not defy combinatorial analysis. Examples of the former approach abound, in fields as diverse as algorithm design [12, 23, 7], program development [4, 22, 19], distributed and parallel computing [6, 5], and artificial intelligence [17, 10, 11]. The latter approach, although not quite as popular as the former, has investigated two broad categories of analytically solvable N-queens relatives—those that reduce constraints [20] and those that increase them [18, 3]. This widespread affinity for the N-queens problem stems, at least in part, from its possession of a property that frequently characterizes difficult problems: global constraints. The moment one queen is placed on the board, the number of possibilities open to *all other* queens is greatly reduced. Constraints with this attribute are generally assumed to be nondecomposable. This paper uses a divide-and-conquer technique which splits the input into distinct subsets to generate a "family of solutions." Since divide-and-conquer algorithms imply efficient parallel and distributed implementations [13], this decomposition of global constraints seems to be significant.

2. Background

The N-queens problem is an example of *constraint satisfaction,* a class of problems that involves assigning values to variables subject to a set of binary constraints [8, 17, 21]. Members of this class are generally solved with backtracking [12], a worst case exponential method which is frequently supported by heuristics [10, 17, 15, 2]. In the N-queens problem, rows can be regarded as variables and columns as values. Given the board size, N, as input, the $\langle$variable, value$\rangle$ pairs can be expressed as a permutation P of 0 to $N-1$, where $P(i)$ is the column of the queen in the ith row. This representation alone is enough to guarantee that no two queens will be in the same row or column, leaving only the diagonal constraints to be verified. Since the diagonals going from top left to bottom right can be characterized by $(i - P(i)) =$ constant, and those from top right to bottom left by $(i + P(i)) =$ constant, P is a solution if and only if for $i \neq j$, $(i - P(i)) \neq (j - P(j))$, and $(i + P(i)) \neq (j + P(j))$.

One interesting extension to the N-queens problem involves placing N *superqueens* on a toroidal board [18]. When a superqueen reaches an edge of the board, it can wrap around to the opposite edge. This power overconstrains the problem by connecting previously separate diagonals; the resulting (toroidal) board has N rows, N columns, and two sets of N diagonals (characterized by $\{(\text{row} - \text{column}) = \text{constant mod } N\}$ and $\{(\text{row} + \text{column}) = \text{constant mod } N\}$), each containing N squares. Pólya [18] showed that an N-superqueens solution exists if and only if N is not divisible by 2 or 3. Since the N-superqueens is an overconstrained variation, any N-superqueens solution solves the N-queens problem as well.

Chandra [3] developed the theory of *independent permutations* which he used to characterize a family of solutions to the N-superqueens problem, the *regular solutions,* in which the permutation P can be characterized by $P(i) = Ai + B \pmod N$. The permutation $P(i) = 2i$, for example, starts with a superqueen in the upper left-hand corner and proceeds around the board placing queens one row down and two columns across. This solution is called the *knight-walk.* Some regular extensions of the toroidal problem are discussed in [3, 14]. As an immediate extension of their existence for any solvable N-superqueens problem, the knight-walk solves the N-queens problem for two-thirds of the odd numbers. Furthermore, removing the top row and leftmost column from the N-by-N board deletes only the queen in the top left-hand corner square. This leaves an $(N-1)$-by-$(N-1)$ board with $(N-1)$ mutually nonattacking queens, solving the $(N-1)$-queens problem, thereby constructing solutions for two-thirds of the even numbers. No other simple board modifications are possible, since removal of any other row and column would either delete two queens or shift the diagonals, or both. The knight-walk and upper left-hand corner removal, then, constitute a solution scheme for two-thirds of all N.

3. The Decomposition Solution to the Superqueens Problem

This section shows how to construct a family of nonregular solutions to the N-superqueens problem, the decomposition solutions. The main idea is to use the factorization of N to apply a divide-and-conquer approach to the problem. Basically, if N can be factored as AB where both A and B are solvable, the N-superqueens problem can be reduced to solving A appropriately chosen copies of the B-superqueens problem.

DEFINITION. Let $N = AB$, where there are solutions to the A-superqueens and B-superqueens problems. A *decomposition solution* breaks the N-by-N board into an A-by-A grid of B-by-B tiles. Tiles corresponding to an A-superqueens solution are filled with a B-superqueens solution; the same B-superqueens solution is used throughout.

DEFINITION. Let $\mathrm{LR}(N)$ refer to the set of diagonals running from top left to bottom right on an N-by-N board. Let $\mathrm{OLR}(N)$ refer to those members of $\mathrm{LR}(N)$ which are occupied by a queen. Similarly, let $\mathrm{WLR}(N)$ refer to the set of N wrapped diagonals on a planar representation of an N-by-N toroidal board running from top left to bottom right. Let $\mathrm{OWLR}(N)$ refer to those members of $\mathrm{WLR}(N)$ which are occupied by a superqueen. Let the diagonals from top right to bottom left be named correspondingly (that is, $\mathrm{RL}(N)$, $\mathrm{ORL}(N)$, $\mathrm{WRL}(N)$, and $\mathrm{OWRL}(N)$).

As an illustration of the manner in which a decomposition solution solves the problem, look at the example in Fig. 1, where $N = 35$, $A = 7$, and $B = 5$. The knight-walk is used as both the 5- and the 7-superqueens solutions. A board set up according to the definition of decomposition clearly contains 35 superqueens, with no two in any one row or column. Furthermore, since the diagonals that result from tiling a plane with 5-by-5 boards are equivalent to those resulting from placing a 5-by-5 board on a torus, the diagonal constraints are met, as well. As shown in the diagram, a diagonal in $\mathrm{WLR}(35)$ passes alternately through members of $\mathrm{LR}(5)$ of lengths 2 and 3. These are exactly the $\mathrm{LR}(5)$ diagonals which combine to form a single element of $\mathrm{WLR}(5)$. The use of a 5-superqueens solution guarantees that only one of these $\mathrm{LR}(5)$ diagonals contains a superqueen. This reduces the realm of possible conflicts to superqueens in the same position on different boards. Examination of the figure reveals that if a member of $\mathrm{WLR}(35)$ passes through corresponding $\mathrm{LR}(5)$ diagonals on two 5-by-5 tiles, the tiles lie along the same $\mathrm{LR}(7)$ diagonal. The use of a 7-superqueens solution precludes the possibility of two such boards being chosen, and thus no two superqueens can be in the same member of $\mathrm{WLR}(35)$. Thanks to the symmetry of the board, an identical argument can be used for $\mathrm{WRL}(35)$.

THEOREM 1. *A decomposition solution solves the N-superqueens problem.*

In order to discuss the general case, some notation must be developed for referring to individual squares on a decomposed board. Each square's location can be specified in two coordinate systems: the N-by-N system and the B-by-B system within the A-by-A grid. In converting between the systems, let $\langle i_N, j_N \rangle$ = tile $\langle i_A, j_A \rangle$ square $\langle i_B, j_B \rangle$, where $i_N, j_N = 0$ to $N - 1$, $i_A, j_A = 0$ to $A - 1$, $i_B, j_B = 0$ to $B - 1$. Since superqueens solutions are used

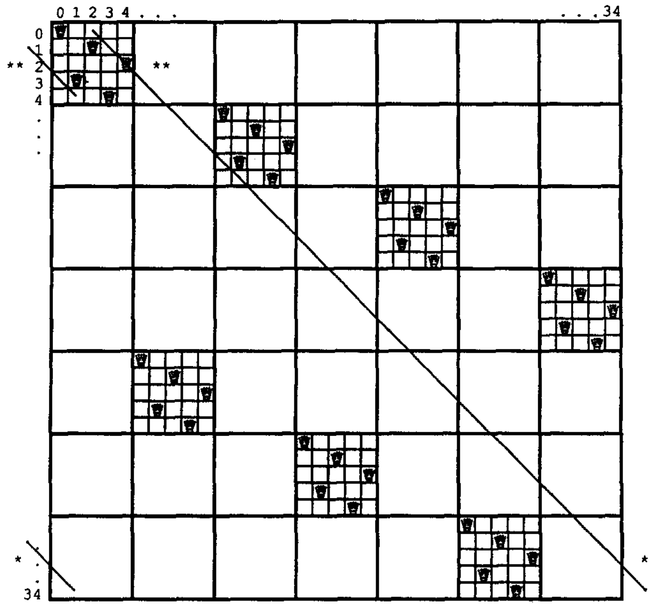


FIG. 1. An example of a decomposition solution. $N = 35$, $A = 7$, and $B = 5$. A diagonal is drawn, along with its continuations in both WLR(B)** and WLR(N)*.

throughout, all boards must be treated as tori, and therefore, all arithmetic is done modulo the subscripted number. The relationship between the systems is given by $i_N = Bi_A + i_B$ and $j_N = Bj_A + j_B$, and two squares *correspond* if they have the same B coordinates. Using this notation, the proof of the theorem becomes a straightforward generalization of the case shown in Fig. 1. More formally:

Proof (Theorem 1). A board configuration is an N-superqueens solution if and only if there are N superqueens on the board, one in each row, column, WLR, and WRL diagonal.

CLAIM 1. *A board set up as specified by the decomposition solution contains exactly N superqueens, one in each row (column).*

Proof (Claim 1). By the definition of decomposition, A copies of the B-superqueens solution are used, totalling $AB = N$ superqueens. These copies are placed in tiles that correspond to an A-superqueens solution, partitioning the rows (columns) into blocks of size B. The B-superqueens solution simply implies an even distribution throughout each block.

CLAIM 2. *Each member of* OWLR(N) (OWRL(N)) *contains one superqueen.*

Proof (Claim 2). There cannot be a conflict between two superqueens on the same tile because any occupied tile contains a B-superqueen solution

The only possible conflicts in OWLR(N), then, are between superqueens on different tiles. In order to prove that conflicts of this nature are impossible as well, consider the LR(B) diagonals that combine to form an OWLR(N) diagonal.

The OLR(B) diagonal of tile $\langle i_A, j_A \rangle$ with a superqueen in square $\langle i_B, j_B \rangle$ is defined by the invariant $(i_B - j_B)$. If $i_B < j_B$, the diagonal runs from the tile's top (square $\langle 0, j_B - i_B \rangle$) to its right side (square $\langle i_B - j_B + B - 1, B - 1 \rangle$). The OWLR$(B)$ diagonal of tile $\langle i_A, j_A \rangle$ containing the OLR(B) diagonal in question is characterized by $(i_B - j_B (\bmod B))$. This OWLR(B) diagonal starts with the OLR(B) diagonal and then wraps around to $\langle i_B - j_B, 0 \rangle$, down to $\langle B - 1, B - 1 + j_B - i_B \rangle$, and back up to $\langle 0, j_B - i_B \rangle$. The OWLR$(N)$ diagonal containing this OLR(B) diagonal takes a course through corresponding squares. Starting with the OLR(B) diagonal, it continues rightward from tile $\langle i_A, j_A \rangle$ square $\langle i_B - j_B + B - 1, B - 1 \rangle$ to tile $\langle i_A, j_A + 1 \rangle$ square $\langle i_B - j_B, 0 \rangle$. This square corresponds to the leftmost square of the wrapped portion of the OWLR(B) diagonal. Thus, the use of a single B-superqueens solution throughout guarantees the absence of superqueens from the portion of the OWLR(N) diagonal in this tile. The OWLR(N) diagonal then continues through tile $\langle i_A, j_A + 1 \rangle$ square $\langle B - 1 + j_B - i_B, B - 1 \rangle$ down to tile $\langle i_A + 1, j_A + 1 \rangle$ square $\langle 0, j_B - i_B \rangle$. This square corresponds to the top square of the original OWLR(B) diagonal. Tile $\langle i_A + 1, j_A + 1 \rangle$, however, lies along the same member of OWLR(A) as $\langle i_A, j_A \rangle$, ensuring that it was not used in the decomposition.

It can easily be shown by induction that this pattern continues. The wrapped diagonal of OWLR(N) alternately passes through tiles disqualified by OWLR(A) and diagonals disqualified by OWLR(B). By symmetry, the same argument holds for $i_B > j_B$. If $i_B = j_B$, the tile diagonal in question is the main diagonal, which passes only through tiles disqualified by the A-superqueens solution. Thus, no member of OWLR(N) contains more than one superqueen. By the pigeonhole principle, OWLR(N) = WLR(N), and symmetrically, OWRL(N) = WRL(N). Thus, decomposition constitutes an N-superqueens solution. Q.E.D.

In addition to defining a nonregular family to solve the N-queens problem, (i.e., the permutation of rows, P, cannot be characterized by $P(i) = Ai + B(\bmod N)$ for any A and B), the decomposition technique may be applied recursively to offer some insight into the enumeration problem. Since every ordered split of N into its (not necessarily prime) factors corresponds to at least one decomposition solution, splitting N as $N = f_1 f_2 \cdots f_k$ allows the first factor to play the role of A and break the N-by-N board into a grid of (N/f_1)-by-(N/f_1) tiles. Any solution to the f_1-superqueens problem can then be used to choose tiles, and the remaining factors then recursively fill the chosen tiles. Thus, if M is a proper divisor of N, any M-superqueens solution gives rise to at least one N-superqueens solution. The existence of at least one N-superqueens solution that is not built up from any M-superqueens solution, namely the knight-walk, guarantees that there are fewer M-superqueens solutions than N-superqueens solutions. This induces a partial order on the number of solutions, namely that if there is a solution to the N-superqueens problem, and M is a proper divisor of N, there are fewer M-superqueens solutions than N-superqueens solutions. Further exploitation of properties of integer sequences and techniques of combinatorial enumeration may be helpful in finding a better lower bound for the problem of enumerating all N-queens solutions.

4. A General Solution to the Queens Problem

This section develops a general solution scheme which connects the toroidal and planar problems. A simple modification of toroidal decomposition yields a family of solutions to the previously unsolved planar boards. The key to this modification lies in the major difference between the knight-walk and decomposition. The knight-walk uses individual superqueens as basic blocks. The interplay between these blocks severely limits the ways in which the solution can be modified. Thus, the knight-walk is of limited use in the construction of a general solution. Decomposition, on the other hand, relies on presolved boards as blocks. The size of these blocks offers a great deal of flexibility in terms of modifications to the solution, allowing decomposition to serve as the infrastructure of a general solution.

Consider a specific type of decomposition, a D-solution, in which the A-superqueens solution contains a superqueen in the upper left hand corner and the B-knight-walk is used. The scheme for constructing a general N-queens solution consists of appropriately modifying a D-solution. This section can be divided into two parts. The first part shows that replacing the top leftmost tile of a D-solution with a smaller tile does not violate any of the problem's constraints. The second part proves that such modifications provide decomposable N-queens solutions for nearly all remaining cases.

Lemma 1. *If there is a solution to N-superqueens and $N = AB$, there is a solution to the $B(A-1)$-queens problem.*

Proof. Consider a D-solution. A simple removal of the top row and leftmost column from the A-superqueens board corresponds to removing the top B rows and B leftmost columns from the N-superqueens board. Just as a solution exists for $(A-1)$-queens, one exists for $(N-B) = B(A-1)$-queens. Q.E.D.

This modification alone is not enough. In order to proceed with the discussion, one more definition is needed:

Definition. For all $P \leqslant N$, call the bottom P rows of the P rightmost columns of an N-by-N board the lower right sub-board of order P.

The notion of a lower right sub-board is important in showing that replacing the top leftmost tile of a D-solution with a smaller tile does not violate any of the problem's constraints. Consider the example of Fig. 2, which shows a 7-by-7 board and its lower right sub-board of order 5. The numbers drawn represent the 5- and 7-knight-walks, respectively. Note that the first three superqueens placed on the 5-by-5 sub-board fall in members of OLR(7). The remainder of the 5-superqueens fall in squares containing 7-superqueens. Thus, OLR(5) is a subset of OLR(7). A similar argument holds in the general case, as well.

LEMMA 2. *Given the knight-walk solutions to the B-queens and C-queens problems (B-knight-walk and C-knight-walk, respectively), $C \leq B$, replacing B's lower right sub-board of order C with the C-queens solution does not add new diagonals to* OLR(B).

Proof. An N-knight-walk solution wraps around the board exactly once, between rows $(N-1)/2$ and $(N+1)/2$. First consider the queens in rows 0 to $(C-1)/2$ of C's coordinates (rows $(B-C)$ to $((C-1)/2+(B-C))$ of B's coordinates). Replacing B's lower right sub-board of order C with the C-knight-walk puts the C-knight-walk's upper left-hand corner on B's main diagonal, in square $\langle (B-C), (B-C) \rangle$, thereby aligning the top leftmost queens of the two solutions. Following the knight-walks from these queens,

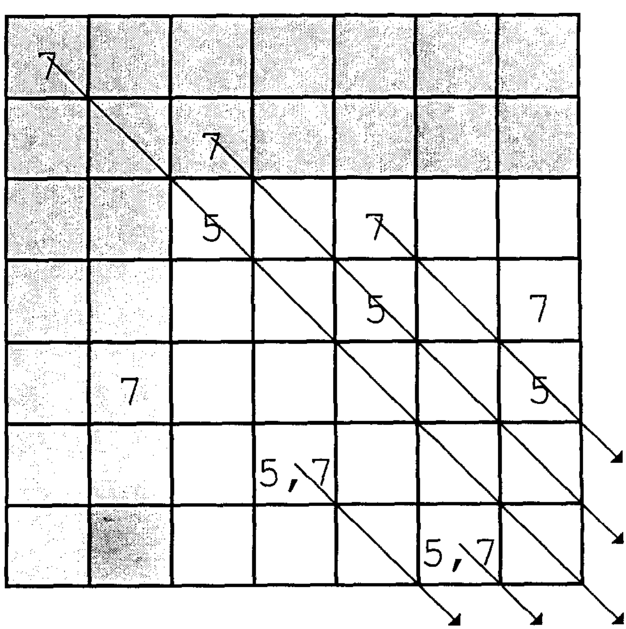


FIG. 2. A 7-by-7 board and its lower right subboard of order 5. The knight-walk solutions have been drawn in.

the ith queen ($i = 0$ to $(C - 1)/2$) is placed on the diagonal characterized by column − row = i (using either coordinate system). This correspondence ensures that each of C's queens placed before the wrap is placed in a member of OLR(B). Next, look at the remaining queens. The first of the C-knight-walk's queens placed after the wrap is in square $\langle (C + 1)/2, 1 \rangle$, which, in B's coordinates, is $\langle ((C + 1)/2 + (B - C)), 1 + (B - C) \rangle$. Similarly, B's wrap lands in square $\langle (B + 1)/2, 1 \rangle$. The pointwise difference between these coordinates is $((B - C)/2, (B - C))$, separating the squares by $(B - C)/2$ knight steps. The size of this separation guarantees that one of B's queens will fall on the square occupied by C's queen. From then on, the two knight-walks will follow the same path. Thus, no diagonals are added to OLR(B).

Q.E.D.

LEMMA 3. *If there are solutions to N-superqueens and C-superqueens, $N = AB$, and $C \leqslant B$, then there is a solution to the $B(A - 1) + C$-queens problem.*

Proof. This follows directly from Lemmas 1 and 2. Set up a D-solution where $N = AB$, and remove the top B rows and leftmost B columns as in Lemma 1. Next, add C new rows and columns and put a C-knight-walk in the new corner board (which corresponds to the lower right sub-board of the removed size-B board). Q.E.D.

These lemmas enlarge the class of board sizes solved by the methods discussed here to all $N = B(A - 1) + C$ for some A, B, C not divisible by 2 or 3, and $C \leqslant B$. Clearly, knight-walk and decomposition are subsumed by this scheme. For a decomposition set $C = B$, and for a knight-walk set $C = B = 1$.

DEFINITION. Let $N = (A - 1)B + C$, where there are solutions to the A-, B-, and C-superqueens problems, and $C \leqslant B$. A *general solution* starts with a D-solution to the AB-superqueens problem. The top B rows and B leftmost columns of the decomposition are replaced with C rows and C columns. The C-knight-walk is then placed in the newly created C-by-C tile in the upper left-hand corner of the board.

The question remains, for which of the remaining board sizes does this general solution work? The following two lemmas provide the answer: (almost) all N. In order to find a general solution there must first be a D-solution that can be modified appropriately. The need for a D-solution, in turn, points to the importance of determining the conditions under which A and B exist such that $B(A - 1) < N < BA$ and then finding an appropriate A and B.

LEMMA 4. *Let N be an odd number divisible by* 3. *The family of pairs of odd numbers A and B, $A = 2$ mod 3 and $B \neq 0$ mod 3, such that $B(A - 1) < N < BA$, gives a family of general solutions to the N-queens problem.*

Proof. Let $N = B(A-1) + C$. Then

— $\{B(A-1) < N < BA\}$ implies $\{0 < C < B\}$.
— $\{A = B = 1 \bmod 2\}$ and $\{N = 1 \bmod 2\}$ imply $\{B(A-1) = 0 \bmod 2\}$, and thus $\{C = 1 \bmod 2\}$.
— $\{A = 2 \bmod 3\}$ and $\{B \neq 0 \bmod 3\}$ imply $\{B(A-1) \neq 0 \bmod 3\}$.
— $\{B(A-1) \neq 0 \bmod 3\}$ and $\{N = 0 \bmod 3\}$ imply $\{C \neq 0 \bmod 3\}$.

In other words, if an appropriate D-solution can be found, a C that meets the requirements of Lemma 3 can be found as well: odd, not divisible by 3, and no larger than B. Q.E.D.

Lemma 4 characterizes the family of general solutions for a given N. In lemma 5, one member of the family is shown to be applied to almost all N.

LEMMA 5. *For odd N divisible by* 3, $N \neq 3, 9, 15, 27, 39$, *setting* $A = 5$ *guarantees the existence of a B that meets the specifications of Lemma* 4.

Proof. This follows directly from the requirements of Lemma 4. According to these specifications, B must be an odd number not divisible by 3, or a member of one of two equivalence classes, (i) $B = 1 \bmod 6$ and (ii) $B = 5 \bmod 6$. Let N be an odd number divisible by 3. Let B_1 be the largest B such that $4B_1 < N$. Since B_1 is well defined only for $N > 3$, the case of $N = 3$ is ruled out of consideration (obviously, there is no solution to the 3-queens problem). If $N < 5B_1$ then B_1 is as required by Lemma 4 (set $C = N - 4B_1$, and A, B, and C are all defined as explained above), so assume $N > 5B_1$. Let B_2 be the smallest B larger than B_1, or the smallest B such that $4B_2 > N$. If B_1 is in equivalence class (i), $B_2 = B_1 + 4$. Otherwise, B_1 is in equivalence class (ii), and $B_2 = B_1 + 2$. In the first case, $5B_1 < N < 4B_2 = 4(B_1 + 4)$, which implies $B_1 \leqslant 14$. The only numbers which satisfy these conditions for B_1 are 1, 7, and 13. In the second case, $5B_1 < N < 4B_2 = 4(B_1 + 2)$, which implies $B_1 \leqslant 6$. The only number which satisfies these conditions for B_1 is 5. The intervals of candidates for N, then, are $5B_1 < N < 4B_2$, where B_1 is one of the aforementioned four numbers. In other words, one of the following is true: $\{5 < N < 20\}$, $\{25 < N < 28\}$, $\{35 < N < 44\}$, or $\{65 < N < 68\}$. The only odd numbers divisible by 3 in these intervals are 9, 15, 27, and 39. Q.E.D.

Although a family of solutions is given by any triple $(A-1, B, C)$ meeting the specifications of Lemma 3, assigning A a value other than 5 does not solve any of the remaining cases. Since Lemma 4 required that $A = 2 \bmod 3$, the next value that A could be assigned is 11. The minimal non-trivial value for B, however, is 5 (recall $B = 1$ yields the knight-walk), so the smallest board size that $A = 11$ could solve is 50. This is larger than the largest unsolved board. As for the hitherto unsolved even numbers, the construction described in the general solution guarantees the existence of a queen in the upper left-hand corner. Removing the top row and leftmost column gives solutions to all even numbers but 2, 8, 14, 26, and 38. As far as these few cases are concerned, the N-queens problem is unsolvable if $N = 2$ or 3, and the others can be looked up in a table.

The results of the solution scheme presented in this section can be summarized as:

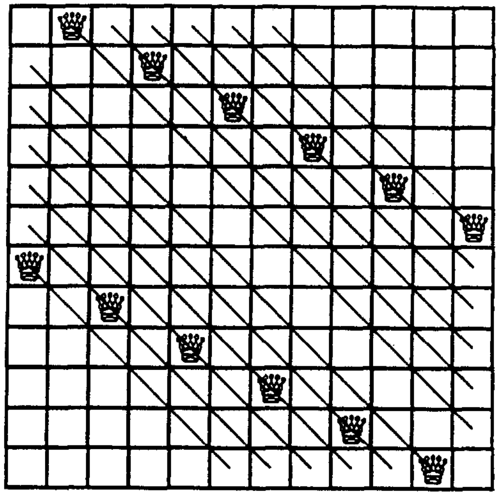
THEOREM 2. *The general scheme yields a nonempty family of solutions to the N-queens problem for all N,* $N \neq 2, 3, 8, 9, 14, 15, 26, 27, 38, 39$.

5. CONCLUSIONS

This paper investigated both the N-queens and the N-superqueens problems, presented new families of solutions to each of them, and clarified the relationship between them. The solutions are based on a divide-and-conquer technique. What seems to be the most important methodological aspect of these results, the decomposability of global constraints, was realizable only through the consideration of a combinatorially simplified (and overconstrained) problem. The extent to which this technique of simultaneously studying two related problems and then generalizing the result found for the special more constrained case to the general case (e.g., from toroidal board to planar board) may be applied to other domains, is unknown. With the advent of distributed and parallel computing and the need for decomposable problems to which fast algorithms can be applied, it is encouraging to discover that global constraints in a problem long thought to require backtracking (anathema to distributed and parallel systems) may be divisible-and-conquerable.

APPENDIX: THE YAGLOMS' SOLUTION

Yaglom and Yaglom described a solution to the N-queens problem for all N in [24]. They concentrated on giving one solution for even board sizes without placing queens in the main diagonal. That vacancy allowed the addition of a row, column, and queen to solve odd-sized boards. For even N of the form $N = 6m$ or $N = 6m + 4$, the setup they describe is shown in Fig. 3a. It is basically the knight-walk of this paper (more accurately, the knight-walk minus the queen in the upper left-hand corner). For board sizes of the form $N = 6m + 2$, however, a totally different setup is needed. They exhibited a pattern which works for these boards. Proceeding rightward from the leftmost column, placing successive queens in the diagonals specified by the following pattern (using the diagonal numbering scheme shown in Fig. 3b), a



b

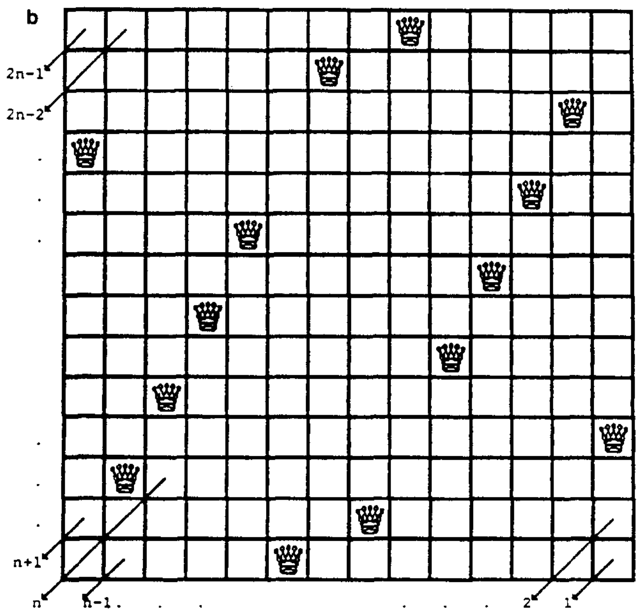


FIG. 3. (a) An example of Yagloms' solution for N of the form $N = 6m$ or $N = 6m + 4$. Here $N = 12$. (b) An example of Yagloms' solution for N of the form $N = 6m + 2$. Here $N = 14$. The diagonal numbers have been drawn in.

solves the problem: $2n - 4, n + 1, n + 2, n + 3, \ldots, 3n/2 - 3, n/2 + 2, 3n/2 - 1, n/2 + 1, 3n/2 - 2, n/2 + 3, n/2 + 4, n/2 + 5, \ldots, n - 1, 4$. The example of $N = 14$ is shown in Fig. 3b.

ACKNOWLEDGMENTS

This research was conducted while both authors were in the Department of Computer Science at Columbia University. Some of the document preparations took place while Bruce Abramson was in residence at the Computer Science Department of UCLA. The authors thank Zvi Galil and Richard Korf for their interest and encouragement and offer special thanks to Ashok Chandra and Martin Gardner for pointing out previous work done on the N-queens problem.

REFERENCES

1. Ball, W. W. R. *Mathematical Recreations and Essays.* Macmillan Co., New York, 1973.
2. Carter, L., Stockmeyer, L., and Wegman, M. The complexity of backtrack searches. *Symposium on Theory of Computing, ACM,* May, 1985, pp. 449–457.
3. Chandra, A. K. Independent permutations, as related to a problem of Moser and a theorem of Póyla. *J. Combin. Theory* **16,** 1 (Jan. 1974), 111–120.
4. Dijkstra, E. W. Notes on structured programming. In Dahl, O. J., Dijkstra, E. W., and Hoare, C. A. R., (Eds.), *Structured Programming.* Academic Press, New York, 1972, pp. 1–82.
5. Filman, R. E., and Friedman, D. P. *Coordinated Computing: Tools and Techniques for Distributed Software.* McGraw–Hill, New York, 1984.
6. Finkel, R., and Manber, U. DIB—A distributed implementation of backtracking. *ACM Trans. Programming Languages Systems* **9** (Apr. 1987), 235–256.
7. Floyd, R. W. Nondeterministic algorithms. *J. Assoc. Comput. Mach.* **14,** 14 (1967), 636–644.
8. Freuder, E. C. A sufficient condition for backtrack-bounded search. *J. Assoc. Comput. Mach.* **32,** 4 (Oct. 1985), 755–761.
9. Gardner, M. *The Unexpected Hanging and Other Mathematical Diversions.* Simon & Schuster, New York, 1969.
10. Gaschnig, J. Performance measurement and analysis of certain search algorithms. Ph.D. thesis, Carnegie-Mellon University, May 1979.
11. Haralick, R. M., and Elliot, G. L. Increasing tree search efficiency for constraint satisfaction problems. *Artificial Intelligence* **14,** 3 (Oct. 1980), 263–313.
12. Horowitz, E., and Sahni, S. *Fundamentals of Computer Algorithms.* Computer Science Press, Rockville, MD, 1978.
13. Horowitz, E., and Zorat, A. Divide-and-conquer for parallel processing. *IEEE Trans. Comput.* **32** (June 1983), 582–585.
14. Hwang, F. K., and Lih, K.-W. Latin squares and superqueens. *J. Combin. Theory, Ser. A* **34,** 1 (Jan. 1983), 110–114.
15. Knuth, D. E. Estimating the efficiency of backtrack programs. *Math. Comp.* **29,** 129 (Jan. 1975), 121–136.
16. Kraitchik, M. *Mathematical Recreation.* Norton, New York, 1942.
17. Pearl, J. *Heuristics.* Addison–Wesley, Reading, MA, 1984.
18. Pólya, G. Uber die 'doppelt-periodischen' Losungen des n-Damen-Problems. In Ahrens, W. (Ed.). *Mathematische Unterhaltungen und Spiele.* Teubner, Leipzig, 1918, pp. 364–374.
19. Schwartz, J. T., Dewar, R. B. K., Dubiner, E., and Schonberg, E. *Programming with Sets: An Introduction to Set 1.* Springer-Verlag, New York, 1986.

20. Wagner, R., and Geist, R. The crippled queen placement problem. Duke University, Jan. 1984.

21. Waltz, D. Understanding line drawings in scenes with shadows. In Winston, P. (Ed.). *The Psychology of Computer Vision.* McGraw-Hill, New York, 1975, pp. 19–91.

22. Wirth, N. Program development by stepwise refinement. *Comm. ACM* **14**, 4 (Apr. 1971), 221–227.

23. Wirth, N. *Algorithms + Data Structures = Programs.* Prentice-Hall, Englewood Cliffs, NJ, 1976.

24. Yaglom, A. M., and Yaglom, I. M. *Challenging Mathematical Problems with Elementary Solutions.* Holden-Day, San Francisco, 1964.

MORDECHAI (MOTI) YUNG received a B.Sc. and M.Sc. in mathematics from Tel Aviv University, and an M.Sc. and Ph.D. in computer science from Columbia University. In 1988 he visited IBM Almaden Research Center and is currently with IBM T.J. Watson Research Center. His research interests include distributed and parallel computing, design of algorithms and their complexity, randomized computation, and communication security and cryptography.

BRUCE ABRAMSON received a B.A. in computer science and mathematics in 1983, an M.S. in computer science in 1985, and a Ph.D. in computer science in 1987, all from Columbia University. While pursuing his doctorate, he spent two years in residence at the University of California at Los Angeles. Since 1987, he has been Assistant Professor of Computer Science at the University of Southern California. Dr. Abramson's current research interests focus on the reformulation of artificial intelligence as a decision science. Since completing his dissertation, which introduced a probabilistic interpretation of game-trees, he has been working on extending his results from games to more realistic settings, including robot planning, financial forecasting, and medical decision making.