Direct Search Methods Make a Comeback

Layne T. Watson and Jian He

Departments of Computer Science and Mathematics

Virginia Polytechnic Institute and State University

Blacksburg, VA 24061-0106 USA

ASTC Tutorial, April 1, 2003

Outline

1. Introduction
2. DIRECT Global Search Algorithm (Jones, Perttunen, and Stuckman, 1993)
   - DIRECT = global + local
   - DIviding RECTangles in action
   - Algorithm description
   - Global convergence property
   - Algorithm modifications
3. Dynamic Implementation
   - Motivation
   - Box structures
   - Linked list structures
   - Performance studies
5. Applications
   - Aircraft design: HSCT (high speed civil transport)
   - Wireless communication system design: S^4W (site-specific system simulator for wireless system design)
6. Conclusion
7. References

Introduction
Optimization problem and category

1. **Problem statement:**

$$\min_{x \in D} f_0(x),$$

$$D = \{x \in D_0 \mid f_j(x) \le 0,\ j = 1, \ldots, J\},$$

where $D_0 = \left\{x \in E^n \mid \ell \le x \le u\right\}$ is a simple box constraint set.

2. **Problem category**

- Discrete domain D.

  Combinatorial problems.

- Continuous domain D.

  Unconstrained (or with simple bound constraints) and constrained problems.

Introduction

Optimization approaches (classified by Taylor's approximation)

1. **Second order:**

   For example, Newton-like methods using the Hessian matrix.

2. **First order:**

   For example, steepest descent methods using the gradient vector.

3. **Zero order:**

- Deterministic search methods:

  For example, direct search methods—pattern search (DIRECT) and simplex-based search (multi-directional search, adaptive directional search, etc.), and branch and bound search methods.

- Nondeterministic (heuristic) search methods:

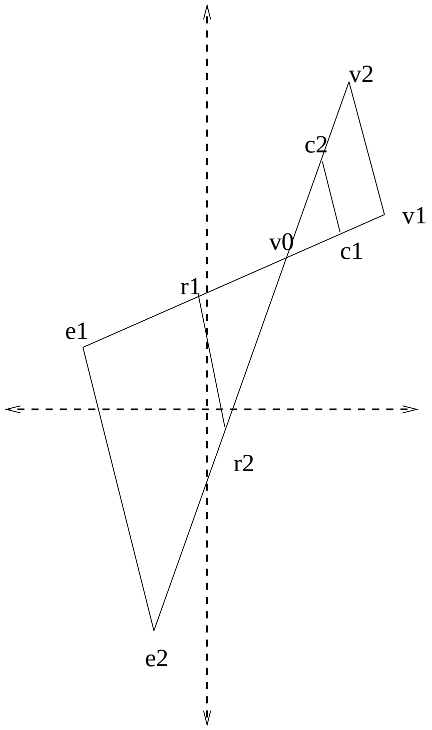
  For example, simulated annealing, Tabu search, and genetic algorithm.

Introduction
Direct search methods

1. **Characteristics**

   Derivative-free; only function values are needed; global convergence properties are guaranteed under some conditions.

2. **Example: simplex-based multi-directional search** (Dennis and Torczon, 1991)



Original simplex: (v0, v1, v2)

Search steps:

(1) Reflection: (v1, v2) to (r1, r2)

(2) Expansion: (r1, r2) to (e1, e2)

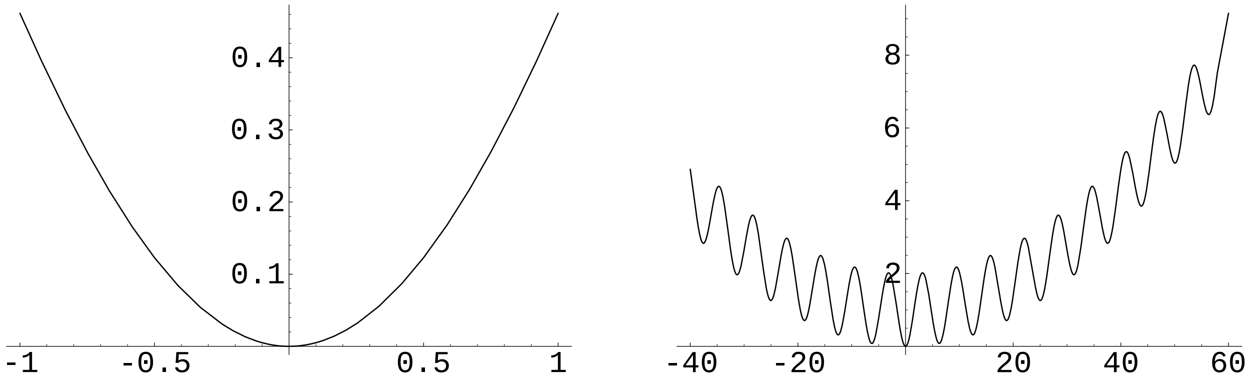
(3) Contraction: (v1, v2) to (c1, c2)

DIRECT Global Search Algorithm

DIRECT = global search + local search

It is a good choice for engineering design problems, which have

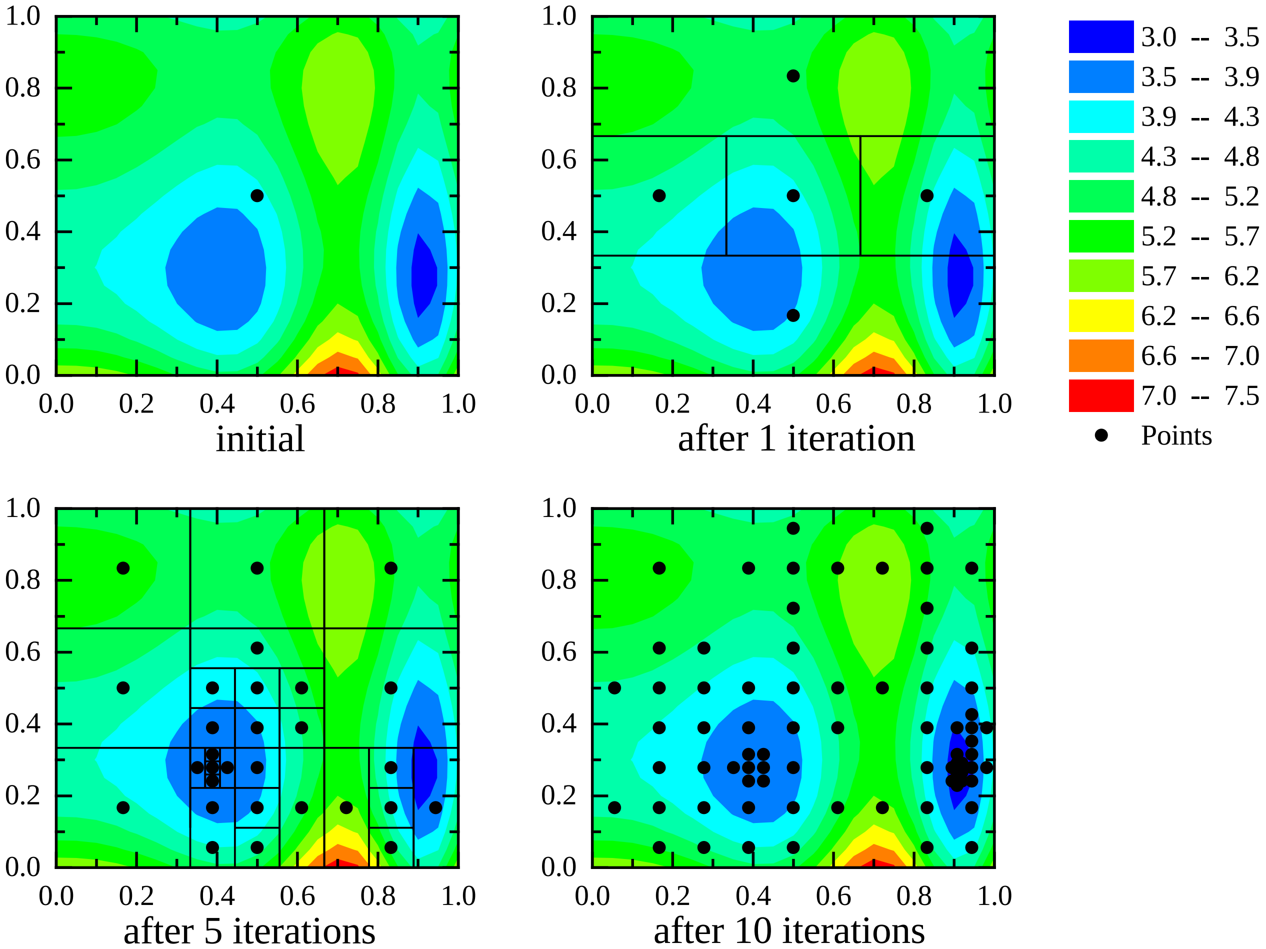
1. no convexity assumptions (local==global for strictly convex functions),
2. multiple local minima (local search is easily trapped),
3. and black-box designs (unpredictable system parameters).



Example: one-dimensional Griewank function with $d = 500$ (right).

$$f(x) = 1 + \sum_{i=1}^{n} \frac{{x_i}^2}{d} - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right)$$

DIRECT Global Search Algorithm
DIviding-RECTangles in action



DIRECT Global Search Algorithm
Algorithm description

Given an objective function f and the design space $D = D_0$:

Step 1. Normalize the design space D to be the unit hypercube. Sample the center point c_i of this hypercube and evaluate $f(c_i)$. Initialize $f_{\min} = f(c_i)$, evaluation counter $m = 1$, and iteration counter $t = 0$.

Step 2. Identify the set S of potentially optimal boxes.

Step 3. Select any box $j \in S$.

Step 4. Divide the box j as follows:

(1) Identify the set I of dimensions with the maximum side length. Let δ equal one-third of this maximum side length.

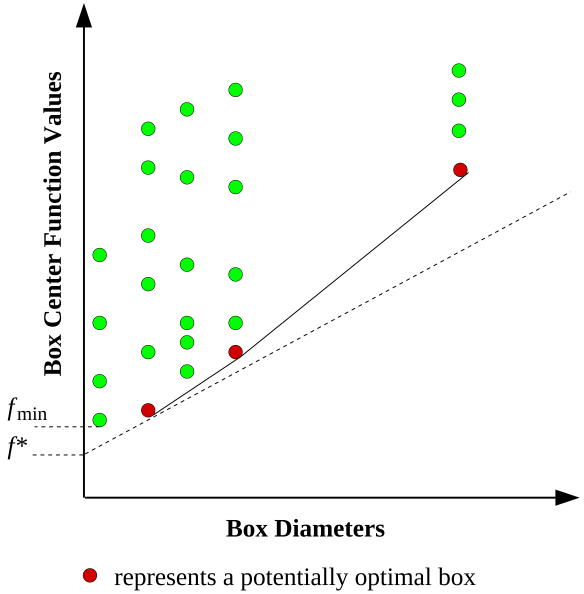
(2) Sample the function at the points $c \pm \delta e_i$ for all $i \in I$, where c is the center of the box and e_i is the ith unit vector.

(3) Divide the box j containing c into thirds along the dimensions in I, starting with the dimension with the lowest value of $w_i = \min\{f(c + \delta e_i), f(c - \delta e_i)\}$, and continuing to the dimension with the highest w_i. Update $f_{\min}$ and m.

Step 5. Set $S = S - \{j\}$. If $S \neq \emptyset$ go to Step 3.

Step 6. Set $t = t + 1$. If iteration limit or evaluation limit has been reached, stop. Otherwise, go to Step 2.

DIRECT Global Search Algorithm
Global convergence property



- Box selection rule: box j is potentially optimal if

$$f(c_j) - \tilde{K} d_j \leq f(c_i) - \tilde{K} d_i,$$

$$f(c_j) - \tilde{K} d_j \leq f_{\min} - \epsilon |f_{\min}|,$$

for some $\tilde{K} > 0$ and $i = 1, \ldots, m$ (the total number of subdivided boxes)

- Lipschitz continuity is required in the domain.

DIRECT Global Search Algorithm
Algorithm modifications from Jones' original DIRECT

1. Box selection rules

- Optional "aggressive switch":

  Switch on/off the convex hull processing used in identifying potentially optimal boxes.

- $\epsilon = 0$ by default.

2. Stopping rules

- Minimum diameter:

  Terminate when the best potentially optimal box's diameter is less than this minimum diameter.

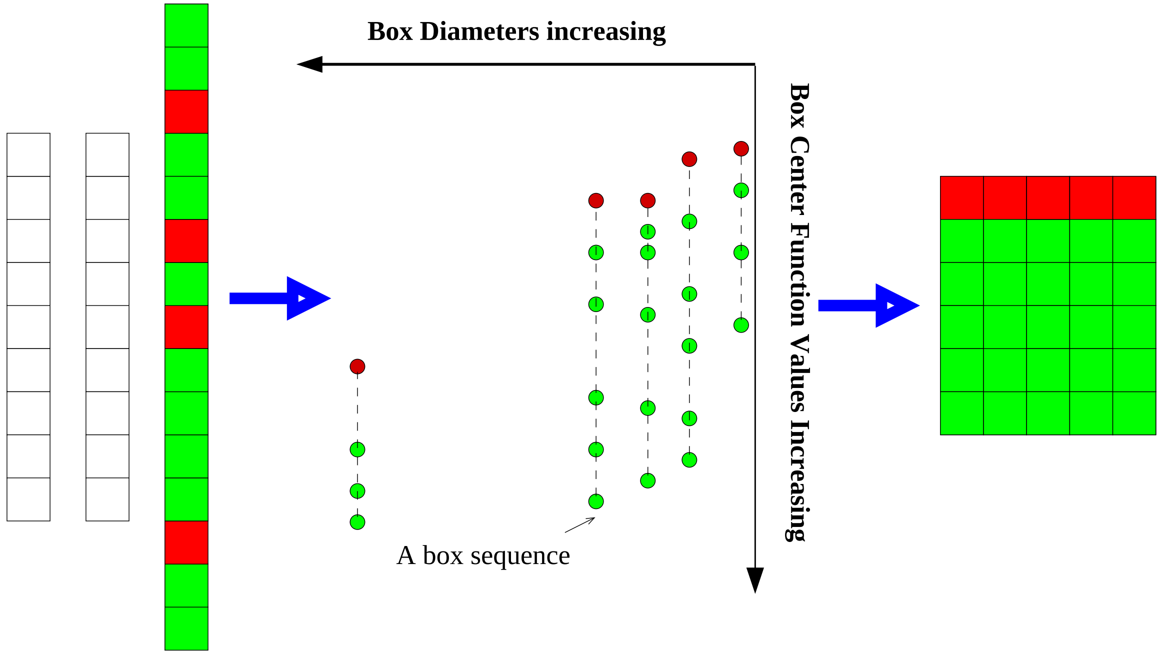
- Objective function convergence tolerance:

$$\tau_f = \frac{\tilde{f}_{\min} - f_{\min}}{1.0 + \tilde{f}_{\min}},$$

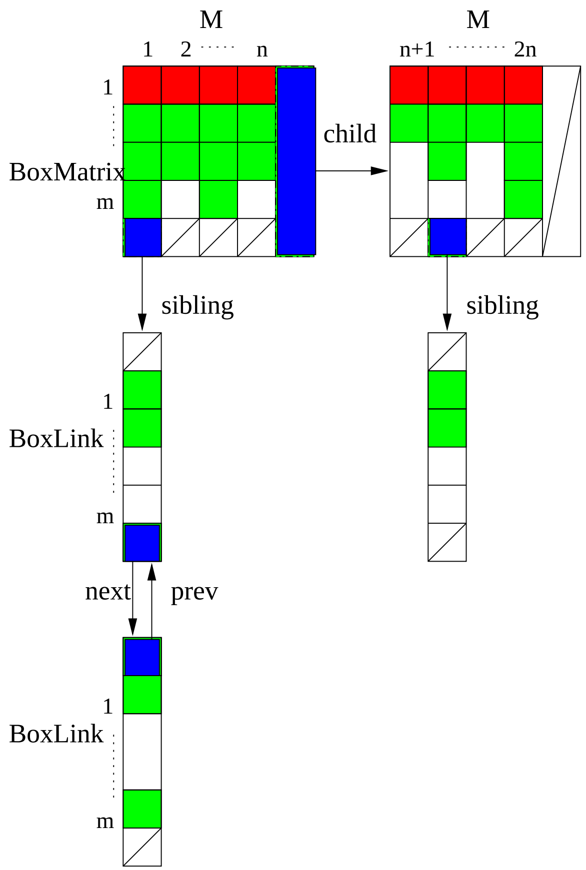
where $\tilde{f}_{\min}$ represents the previous computed minimum. The algorithm stops when τ_f becomes less than a user specified value.

Dynamic Implementation
Motivation

1. **Challenges:**
   - Unpredictable storage requirement
   - Execution overhead of subdividing potentially optimal boxes
2. **Better mapping:** from 1-D data structures to a 2-D data structure.

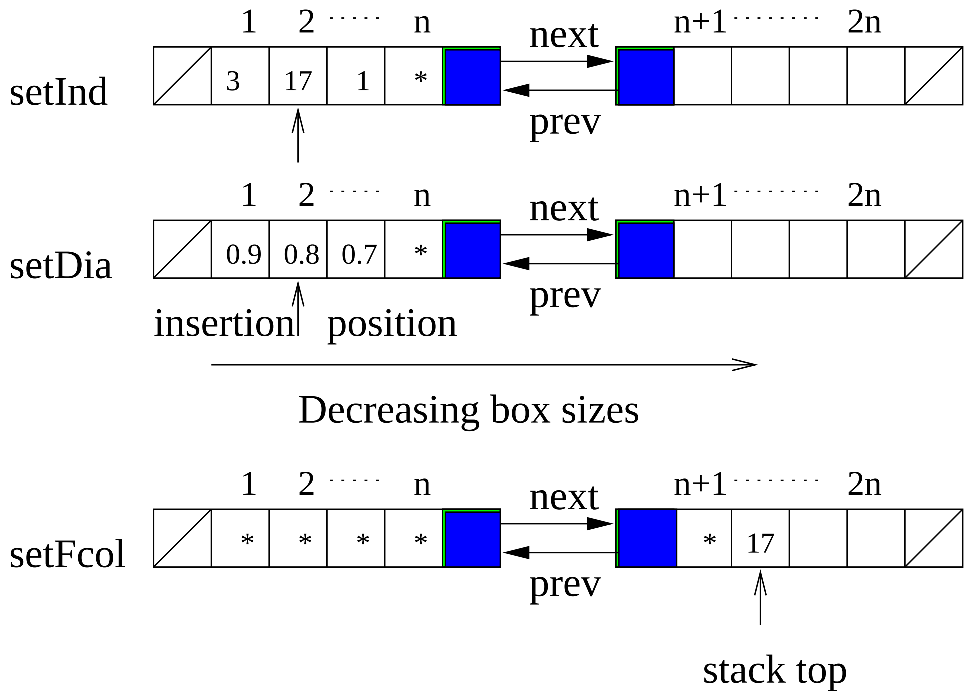


Dynamic Implementation
Box structures



- Two-dimensional dynamic structure
- Priority queue vs. sorted list

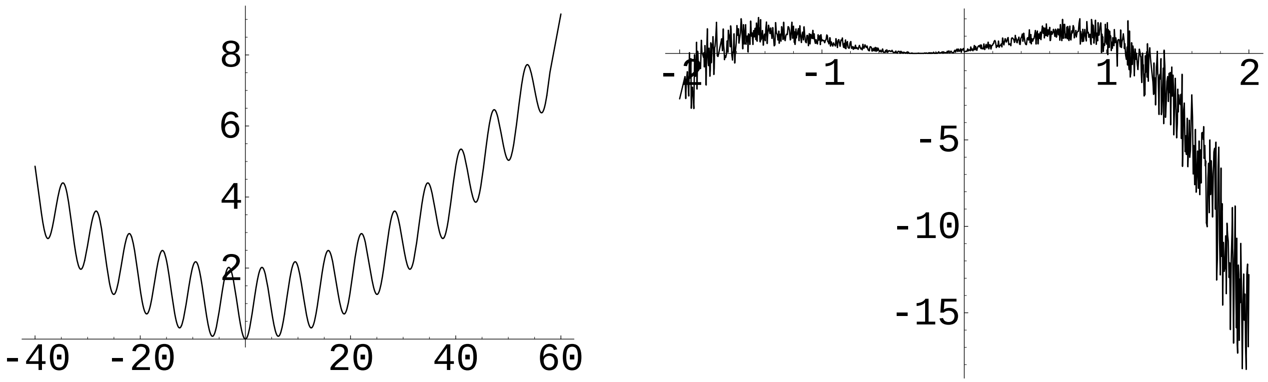
Dynamic Implementation
Linked list structures



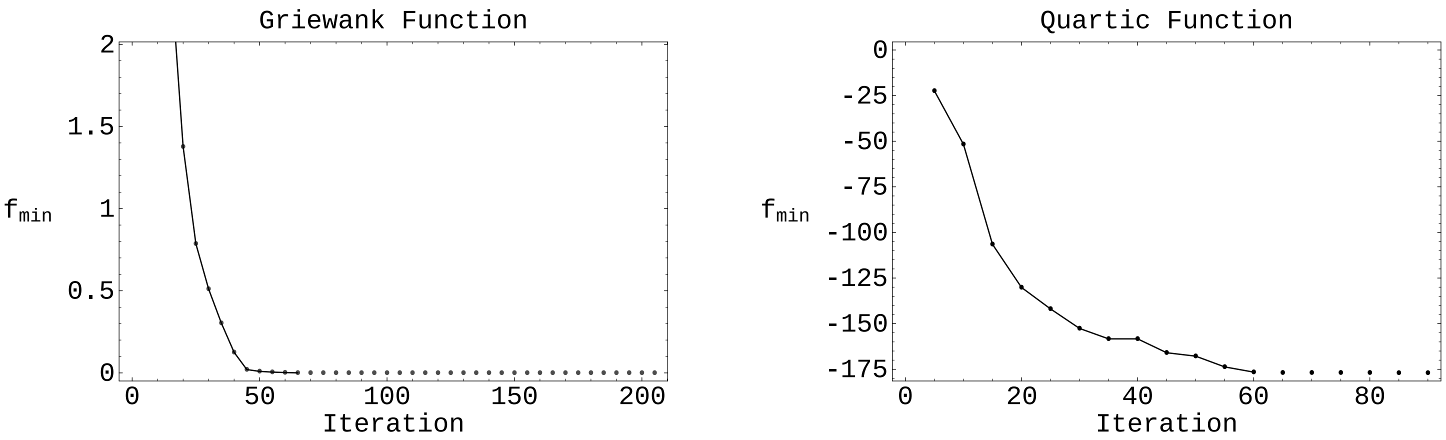
- Maintain 2-D structure
- Recycle box sequence columns

Dynamic Implementation
Performance studies

- Objective function convergence tolerance τ



One-dimensional Griewank function and quartic function



Comparison of $\tau = 0.0001$ (solid) and $\tau = 0$ (dotted)

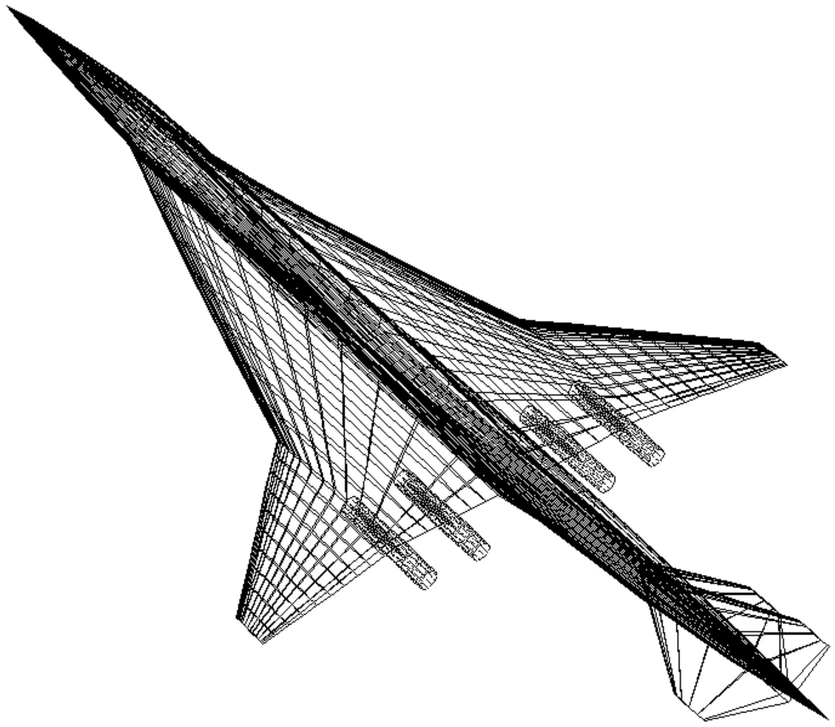
Dynamic Implementation
Performance studies (cont.)

- Comparison of static and dynamic implementations (dimension n, L iterations, time in $msec$, memory usage (maximum working set size) in pages (1 page = 512 bytes)).

			Baker		Gablonsky		dynamic structures	
Problem	n	L	time	memory	time	memory	time	memory
Griewank	2	50	172	10264	34	2224	85	1040
Griewank	5	50	199	11504	34	2352	73	1024
Griewank	10	50	310	15424	51	2648	110	1616
Griewank	15	50	639	18280	88	3232	192	2744
Griewank	20	50	*	*	170	4464	397	6080
Griewank	50	70	*	*	*	*	6161	82664
Quartic	2	50	108	10240	26	2176	25	520
Quartic	5	50	151	11488	31	2240	27	528
Quartic	10	50	441	15432	36	2472	58	1160
Quartic	15	50	1260	18336	54	2992	125	2176
Quartic	20	50	*	*	82	3872	240	4560
Quartic	50	90	*	*	*	*	6572	86656

Application in Aircraft Design: HSCT
Problem scenario

Optimization objective: minimize takeoff gross weight (TOGW) for a range of 5500 nautical miles and a cruise Mach number of 2.4, while carrying 251 passengers.



Typical high speed civil transport (HSCT) configuration.

Application in Aircraft Design: HSCT
Optimization design variables and constraints

1. 28 design variables:

- Geometry of the aircraft: 24 variables in 6 categories
  wing planform,
  airfoil shape,
  tail areas,
  nacelle placement,
  and fuselage shape.
- Idealized cruise mission: 4 variables
  mission fuel,
  engine thrust,
  initial cruise altitude,
  and constant climb rate.

2. 68 constraints in 3 categories:

- Geometry
- Performance
- Aerodynamic

Application in Aircraft Design: HSCT
Problem formulation

1. **Issue #1 — convergence determination.**

   For this study, the algorithm was run for a fixed number of loops or iterations. Since the purpose of the optimization was to identify promising regions of the design space, it was unnecessary to tightly converge to a global optimum.

2. **Issue #2 — incorporation of constraints.**

   Constraints were accounted for through the use of a simple penalty function, as follows. Let x be the 28-dimensional design vector, $f(x)$ the TOGW, and $g_i(x) \leq 0$ the constraints. The constrained optimization problem

   $$\min f(x) \quad \text{subject to } g_i(x) \leq 0,\ i = 1, ..., 68,$$

   is converted to the unconstrained optimization problem

   $$\min f(x) + 10 \sum_{i=1}^{68} \max\{0, g_i(x)\}.$$

Application in Aircraft Design: HSCT
Parallel DIRECT

1. Challenges

- Load balancing: moving from static to dynamic.
- Communication bottleneck: adapting bin size.

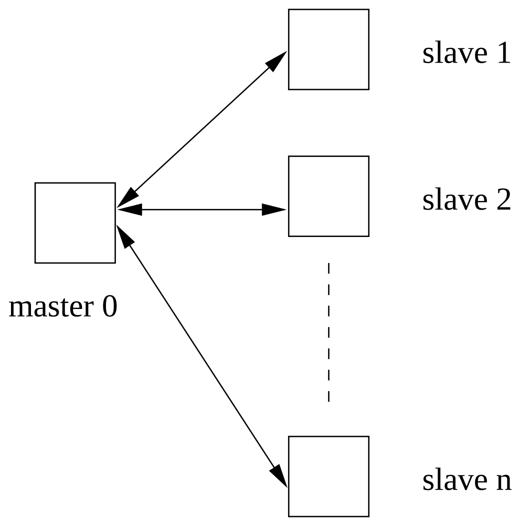
2. Parallel schemes

- Master-slave

  STATIC: static load balancing.

  DLBMS01: dynamic load balancing with bin size 1.

  DLBMS10: dynamic load balancing with bin size 10.



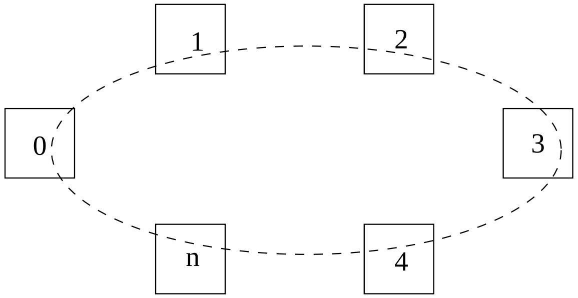
Application in Aircraft Design: HSCT

Parallel DIRECT (cont.)

- Distributed control

  DLBDC: dynamic load balancing with fully distributed control.

  DLBDCT: dynamic load balancing with fully distributed control using pthreads.

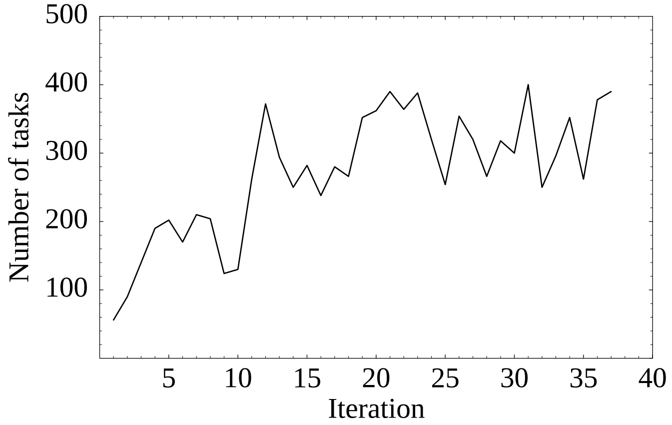


- Aggressive DIRECT

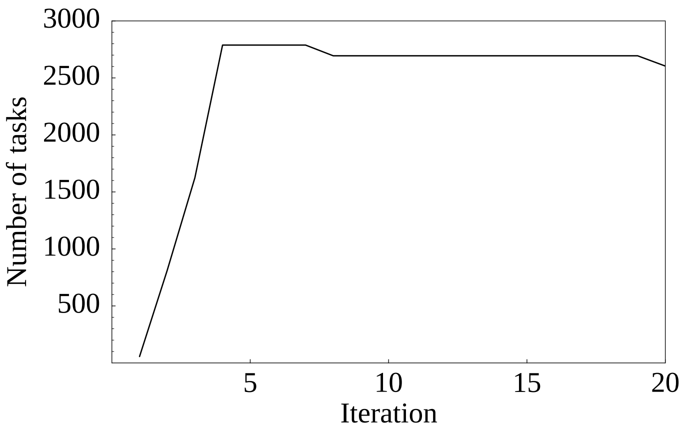
  Discard the step of identifying potentially optimal box set. Each box with the smallest objective function value for that box size is considered “potentially optimal”. This results in a much larger set of new tasks to be evaluated and load balanced at each iteration.

Application in Aircraft Design: HSCT

Task history comparison

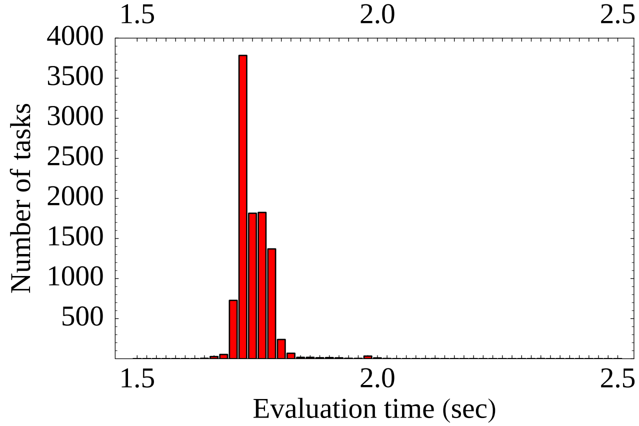


Original DIRECT.

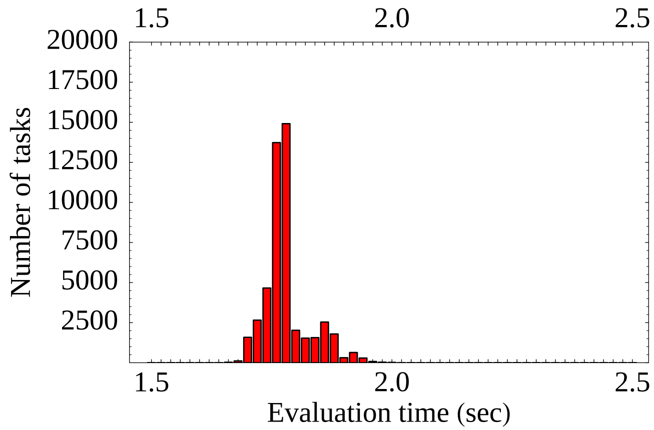


Aggressive DIRECT.

Application in Aircraft Design: HSCT
Time distribution comparison

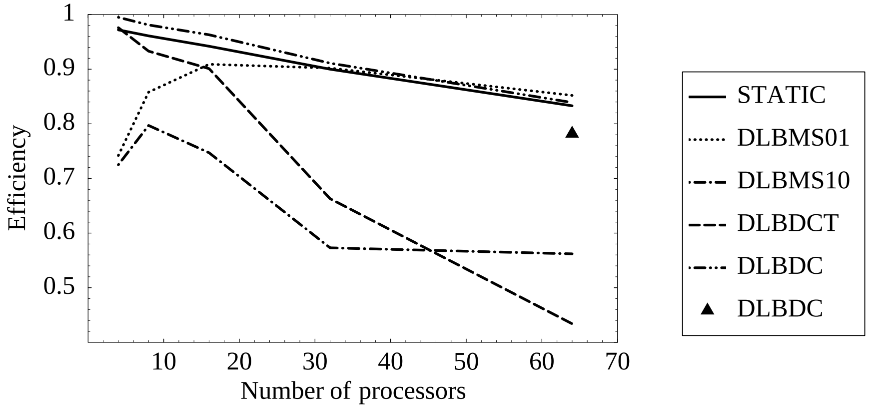


Original DIRECT.

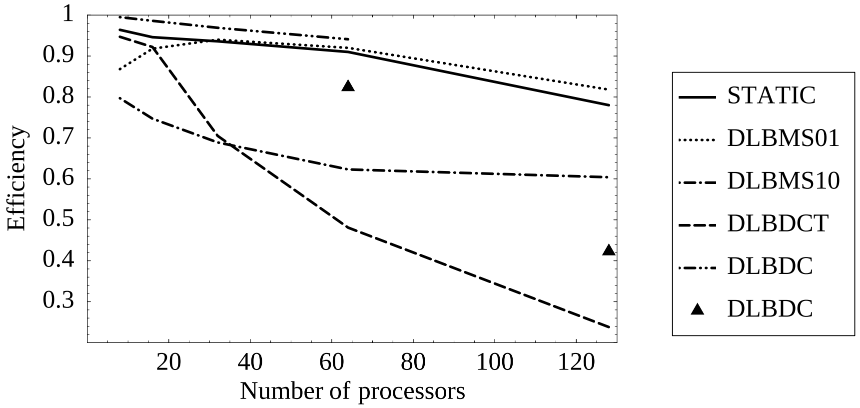


Aggressive DIRECT.

Application in Aircraft Design: HSCT
Parallel efficiency comparison



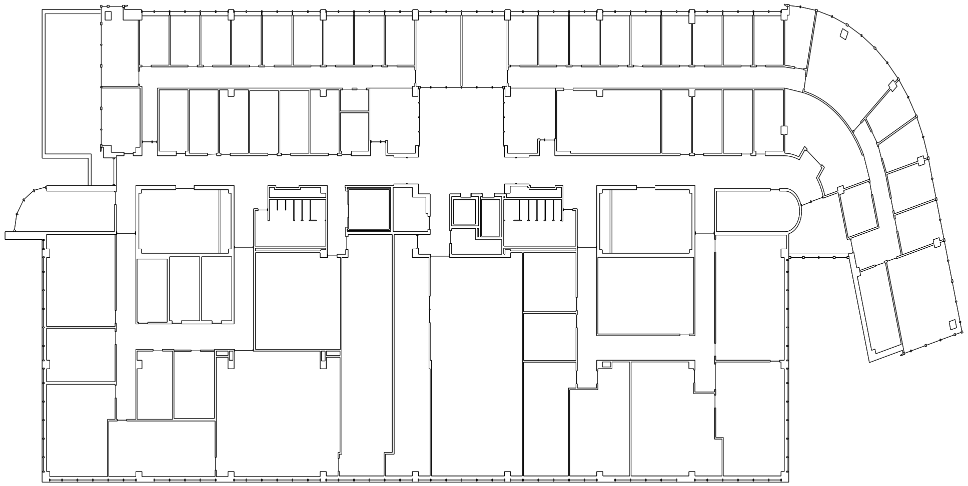
Original DIRECT.



Aggressive DIRECT.

Application in Wireless Design: S^4W
Problem scenario

1. Transmitter placement optimization: ensuring an acceptable level (threshold) of wireless system performance within a geographical area of interest at a minimum cost.



Durham Hall 4th floor, Virginia Tech

2. Problem abstraction:

$$\min_{x \in D} f_0(x),$$

$$D = \{x \in D_0 \mid f_j(x) \leq 0,\ j = 1, \ldots, J\},$$

where $D_0 = \left\{x \in E^n \mid \ell \leq x \leq u\right\}$ is a simple box constraint set.

Application in Wireless Design: S^4W
Objective formulation

1. **Power coverage:**

$$\frac{\text{Number of receivers with received power above threshold}}{\text{Total number of receivers}}$$

2. **Bit error rate (BER):**

$$\frac{\text{Number of incorrectly received bits}}{\text{Total Number of received bits}}$$

3. **Observation:** Discrete vs. continuous.

4. **Reformulation:**

- **Decision variables for n transmitters over m receivers:**

$$X = (x_1, y_1, z_1, x_2, y_2, z_2, \ldots, x_n, y_n, z_n),$$

where $z_i = z_0$.

Application in Wireless Design: S^4W

Objective formulation (cont.)

- **Objective function**

  Average shortfall of the estimated performance metric from the given threshold T:

$$
f(X) = \begin{cases} \dfrac{1}{m}\displaystyle\sum_{i=1}^{m}(T - p_{ki})_+, & \text{coverage,} \\ \dfrac{1}{m}\displaystyle\sum_{i=1}^{m}(p_{ki} - T)_+, & \text{BER.} \end{cases}
$$

p_{ki}: performance metric of transmitter (k,i) evaluated at the ith receiver location, where transmitter (k, i), located at (x_k, y_k, z_0), $1 \le k \le n$, generates the highest power level $P_{ki}(x_k, y_k, z_0) \ge P_{ji}(x_j, y_j, z_0)$, $1 \le j \le n$, at the receiver location i, $1 \le i \le m$.

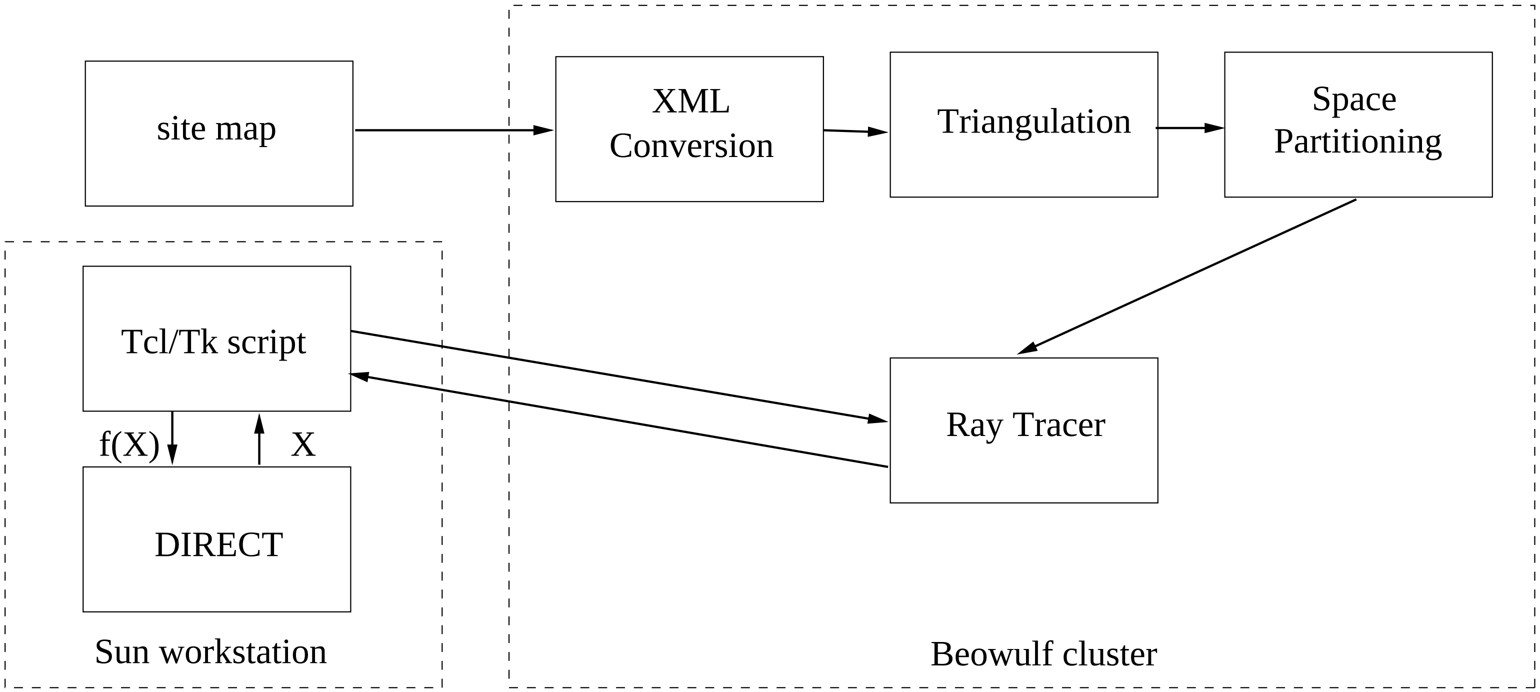
Power coverage optimization:

$p_{ki} = P_{ki}(x_k, y_k, z_0), (T - p_{ki})_+$ is the penalty for a low power level.

BER optimization:

$p_{ki} = \log_{10}(\mathrm{BER}_{ki}), (p_{ki} - T)_+$ is the penalty for a high bit error rate.

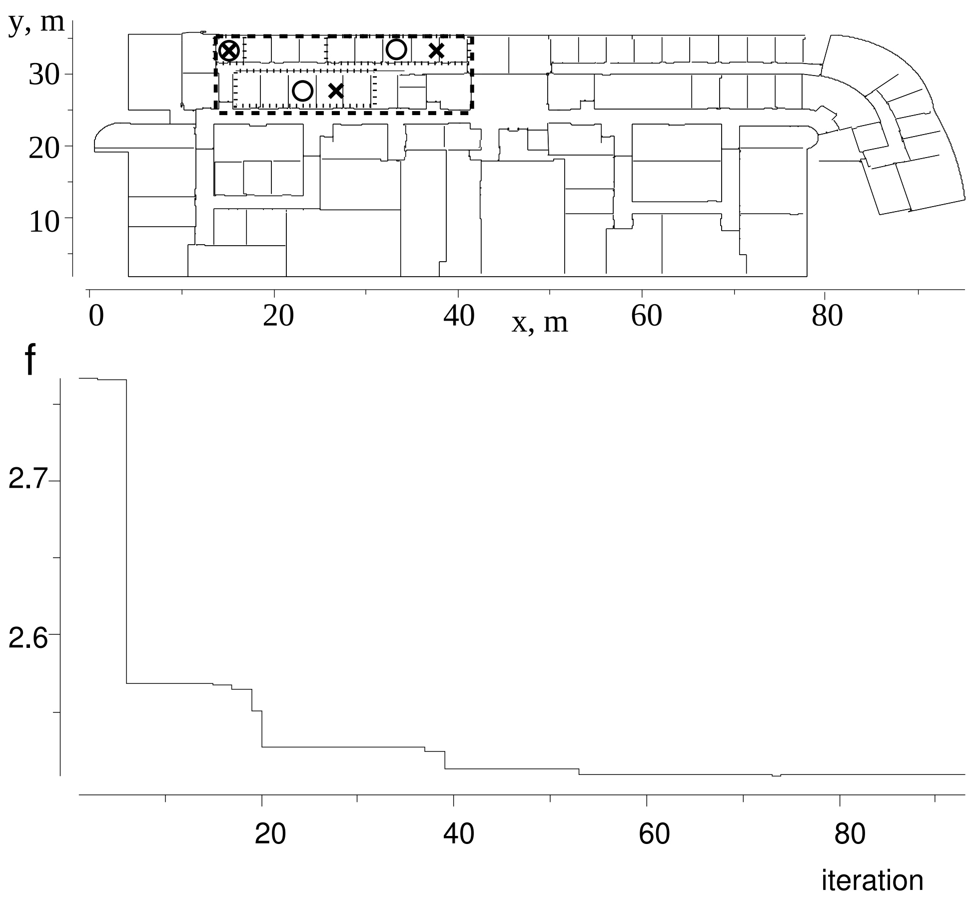
Application in Wireless Design: S^4W
Problem solving environment



- Problem decomposition
- Parallel processing
- Interprocess communication
- Surrogate modeling

Application in Wireless Design: S^4W

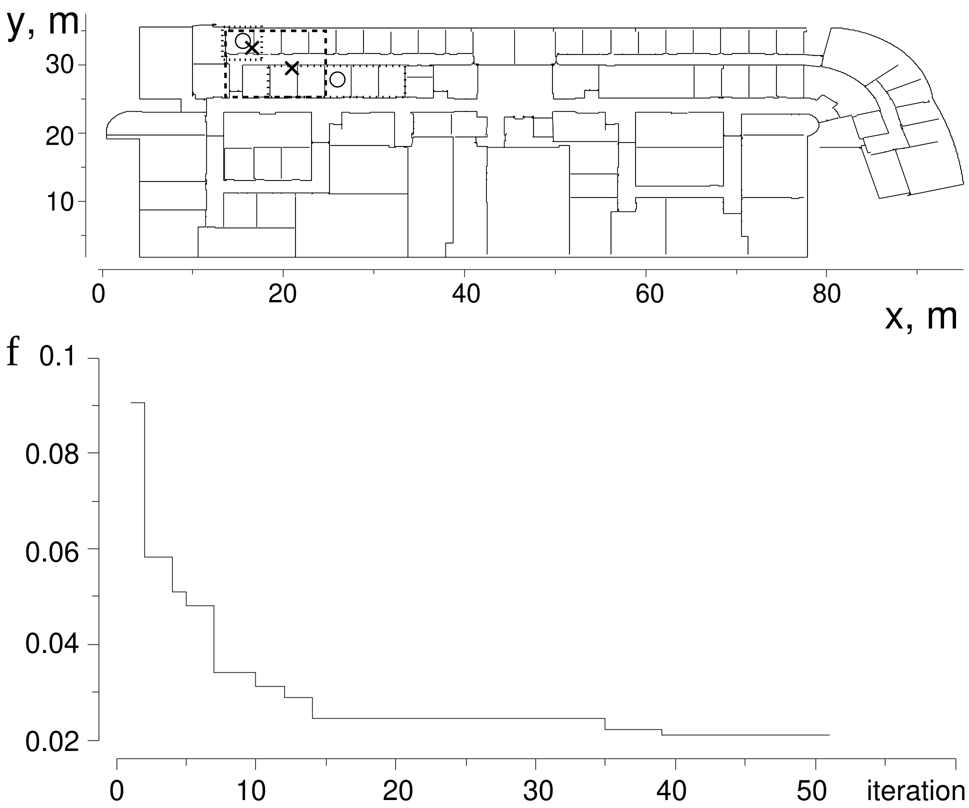
Optimization results



Power coverage optimization results for three transmitters. The starting (optimal) locations are marked with circles (crosses).

Application in Wireless Design: S^4W

Optimization results



BER optimization results for two transmitters. The starting (optimal) locations are marked with circles (crosses).

Conclusion

1. **Effectiveness of DIRECT**

   The DIRECT algorithm solves global optimization problems effectively, especially for continuous unconstrained problems. However, future work is needed for solving discrete problems and nonlinear constrained problems.

2. **Dynamic implementation**

   Addresses the issues of

- data structure extensibility,
- memory allocation efficiency,
- and computation simplicity.

3. **Parallel implementation**

   Unresolved problems include

- load imbalance,
- and communication bottleneck.

References

1. Tutorial URL: http://www.cs.vt.edu/~ltw/ASTC03.ps.gz
2. J. E. Dennis and V. Torczon, "Direct search methods on parallel machines", *SIAM Journal on Optimization*, vol. 1, pp. 448–474, 1991.
3. J. He, A. Verstak, L. T. Watson, C. A. Stinson, N. Ramakrishnan, C. A. Shaffer, T. S. Rappaport, C. R. Anderson, K. Bae, J. Jiang, and W. H. Tranter, "Globally optimal transmitter placement for indoor wireless communication systems", *communicated to IEEE Transactions on Wireless Communications*, 2002.
4. J. He, L. T. Watson, N. Ramakrishnan, C. A. Shaffer, A. Verstak, J. Jiang, K. Bae, and W. H. Tranter, "Dynamic data structures for a direct search algorithm", *Computational Optimization and Applications*, vol. 23, pp. 5–25, 2002.
5. D. R. Jones, C. D. Perttunen, and B. E. Stuckman, "Lipschitzian optimization without the Lipschitz constant", *Journal of Optimization Theory and Application*, vol. 79, no. 1, pp. 157–181, 1993.
6. R. M. Lewis, V. Torczon, and M. W. Trosset, "Direct search methods: then and now", *Journal of Computational and Applied Mathematics*, vol. 124, pp. 191–207, 2000.
7. L. T. Watson and C. A. Baker, "A fully-distributed parallel global search algorithm", *Engineering Computations*, vol. 18, no. 1/2, pp. 155–169, 2001.
8. VTDIRECT URL: http://www.cs.vt.edu/~ltw/VTdirect.ar.gz