Parallelizing and Optimizing the Held-Karp Algorithm for Hamiltonian Circuits

Erik Burton

CME 323: Distributed Algorithms and Optimization

The Held-Karp Algorithm:

Graph = (V,E).
Adjacency Matrix A.
A[i,j] = 1 iff (i,h) ∊ Edges.

Algorithm:
Base Case: if S = {c},
then D(S, c) = A[1,c].

Recursive step:
$D(S, c) = \min_{x \in S-c} (D(S - c, x) + A[x,c])$

Initial call:
$M = \min_{c \in \{2,\ldots,N\}} (D(\{2, \ldots, N\}, c) + A[c,1])$

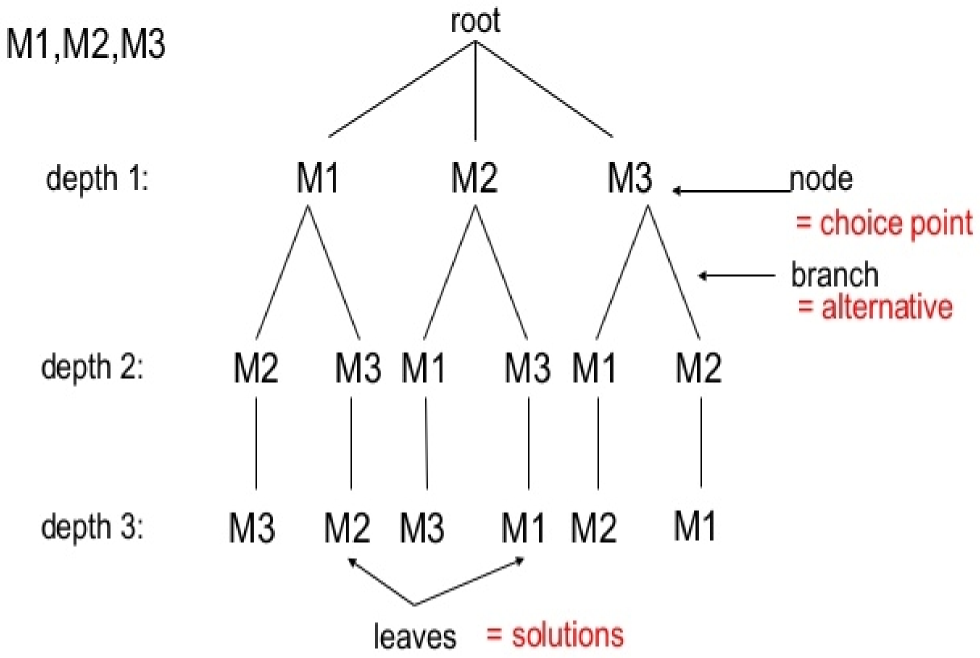


IMAGE: https://image.slidesharecdn.com/tsp-150110102523-conversion-gate01/95/helgauns-algorithm-for-the-tsp-41-638.jpg?cb=1420885662

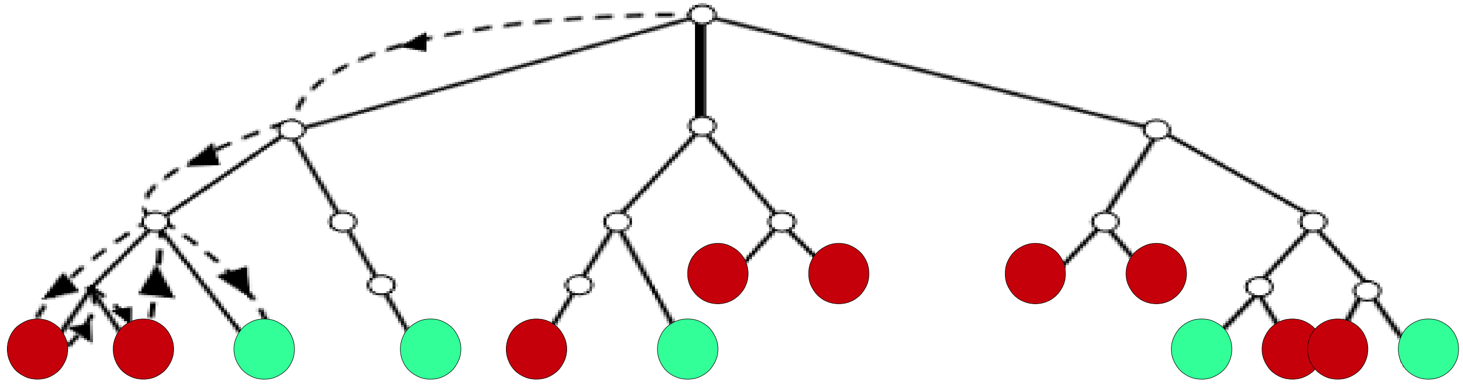
Parallel and Optimized:

Challenges in Parallel:

– Pursuing promising paths (not going broad)
– Avoiding duplicate work
– Retain efficiency, O(1) per processor per step.
– Approach optimality with a reasonable #processors

Solutions:

– Last in – First out Queue behavior (similar to depth first search)
– Efficient backtracking, and noting attempted paths (requires arbitrary CRCW machine)
– Groups of processors that work cooperatively



http://www.universalteacherpublications.com/univ/ebooks/ai/Ch7/Images/depthfirst.gif

Result Overview:

Algorithm	Work (worst)	Depth (worst)	E[Work][I]	E[Depth][I]	Size (worst)	E[Size]	#Processors needed[I]
Held-Karp (1962)	$O(n2^n)$	NA	$\Omega(n^2)$ [III]	NA	$O(n2^n)$	$\Omega(n^2)$ [III]	NA
McKensie-Stout (1993)	$O(n2^n)$	$O(n2^n)$	$O(n)$	$O(\log^*(n))$	$O(n2^n)$	$\Omega(n+\|A\|)$	$O(n/\log^*(n))$
Parallel-Queue (today)	$O(n2^n)$	$O(n)$[II]	$O(Pn)$	$O(n)$	$O(n2^n)$	$O(n+\|A\|)$	$O(\sqrt{n})$

[I] Expected values given the number of processors in the last column.

[III] Lower Bound when $p < ½$ when cycle exists. Depends on p; haven't finalized proof.

[II] Depth running a worst case graph with infinite available processors