Parallel Block Signal Processing in High Speed Wireless Communication Systems

Klaus Hueske [#1], Christian Vincent Sinn [*2], Jürgen Götze [#3]

[#] *Information Processing Lab, University of Dortmund*
Otto-Hahn-Str. 4, 44221 Dortmund, Germany
[1] klaus.hueske@uni-dortmund.de
[3] juergen.goetze@uni-dortmund.de

[*] *Telecommunications Laboratory, University of Sydney*
Sydney, Australia
[2] cvsinn@ee.usyd.edu.au

***Abstract*— Block transmission systems are used for high data rate transmissions over wireless channels. The block structure can be obtained by inserting guard periods, which are used to avoid inter block interference between the transmitted data blocks. However, these guard periods produce a transmission overhead and therefore reduce the throughput of the transmission system. By using so-called overlapping techniques, a block processing structure can be realized without inserting guard periods. In this paper we will use these techniques for two major processing tasks, the data estimation and the error correction Viterbi decoding. By avoiding the overhead in terms of guard periods and initialization/termination sequences for convolutional coding, the throughput of the system can be increased significantly, while the bit error performance is only slightly decreased. Furthermore the overlapping approaches enable a block-wise parallel implementation of the receiver's processing tasks. Depending on the used system parameters (block length, interleaving length, decoding block length), different levels of parallelization can be achieved.**

I. Introduction

Block transmission systems can be used to enable high data rate transmissions over a dispersive wireless channel with reasonable computational complexity. The block structure can be obtained by using guard periods (e.g. cyclic prefixes), which prevent inter block interference (IBI) caused by the channel convolution [1].

Guard periods can be used to remove IBI, but they also cause a reduction of the achievable data throughput, as no information payload can be transferred during their transmission. For example in the WLAN standard [2], 20 % of the total transmission time is allocated to guard periods.

To countervail this overhead the block length could be increased, which leads to several drawbacks: The common assumption that the channel is time invariant during the estimation of one data block becomes unrealistic if the block size is too large. Furthermore the computational complexity, storage requirements and processing delay will be increased, which is a major drawback for real time mobile communication systems.

Using an overlapping estimation scheme [3], we can combine the advantage of short block lengths and avoidance of guard periods. This is realized by detecting the received data block-wise, accepting that the IBI will corrupt the estimated data. By further analysis one can find that the data is mostly corrupted in the beginning and the end of one block, leading to an error distribution with a bathtub like shape. This fact can be exploited by using only the middle, less corrupted part of the detected block, while omitting the more erroneous outer parts. Applying this to several independent overlapping blocks will enable an efficient block-wise data estimation with only slightly decreased bit error performance.

A similar block processing structure can be found for convolutional channel codes, where the blocks are separated using so-called initialization and termination sequences. As these codes are also based on convolution, the same overlapping techniques can be applied for the decoding process [4].

Focusing on implementation aspects one can see that all overlapping blocks can be processed independently, i.e. a fully parallel implementation of the signal processing tasks is feasible. Depending on the used hardware architecture and the used system parameters (e.g. block length, interleaving length,...) different levels of parallelization are possible. In this paper we will present three different parallelization methods, involving the processing blocks of data estimation, deinterleaving and channel decoding.

The paper is organized as follows: In Section II we will first describe the underlying data model, which is used to derive the overlapping estimation scheme and the overlapping decoding scheme. The performance of these approaches in terms of bit error rate (BER) is evaluated in Section III. Three different implementation methods are presented in section IV and final conclusions are drawn in Section V.

II. Overlapping Block Processing

A. Data Model

The model describes the transmission of a data vector $\boldsymbol{d} \in \mathbb{C}^V$ of length V, over a time dispersive wireless channel, which is described by its discrete impulse response $\boldsymbol{h} \in \mathbb{C}^L$ of length L. The channel is assumed to be time invariant during the transmission of the data vector $\boldsymbol{d}$. Noise effects are considered by adding a noise vector $\boldsymbol{n}$, which is obtained by sampling a white Gaussian noise process with power σ^2. The

received vector $\boldsymbol{x} \in \mathbb{C}^{V+L-1}$ can be computed by convolution of the data vector $\boldsymbol{d}$ with the impulse response $\boldsymbol{h}$. By using the channel convolution matrix $\boldsymbol{H} \in \mathbb{C}^{(V+L-1)\times(V)}$ the model can be summarized in

$$\boldsymbol{x} = \boldsymbol{H}\boldsymbol{d} + \boldsymbol{n}. \tag{1}$$

The receiver has to compute an estimate $\hat{\boldsymbol{d}}$ of the transmitted data $\boldsymbol{d}$. It is assumed that the received vector $\boldsymbol{x}$, the channel impulse response and an estimate of the noise power σ^2 are known at the receiver. The transmitted data is uncorrelated and has a mean power of $\sigma_d^2 = 1$, therefore the MMSE (Minimum Mean Square Error) estimate of the transmitted data is given as

$$\hat{\boldsymbol{d}}_{MMSE} = (\boldsymbol{H}^H\boldsymbol{H} + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{H}^H\boldsymbol{x}. \tag{2}$$

Calculating the estimate $\hat{\boldsymbol{d}}$ for long data vectors directly is hardly feasible for high speed wireless communication systems, due to the high demand on computing power, storage requirements and large processing delay time. A common way to solve this problem is breaking down the large matrix $\boldsymbol{H}$ into independent block matrices $\boldsymbol{H}_B$ by using guard periods. To make these submatrices independent, i.e. to avoid IBI, the guard periods of length $L-1$ have to be inserted periodically after each data block. Assuming a zero noise level, the transmitted data can then be recovered without error.
Different kinds of guard periods like Zero Padding (ZP), Known Symbol Padding (KSP) and Cyclic Prefixing (CP) are known [5]. The latter is especially known in the context of CP-OFDM systems, as it allows to calculate the EVD (Eigenvalue Decomposition) of the submatrices $\boldsymbol{H}_B$ efficiently using the FFT (Fast Fourier Transform) [6]. In this paper a CP-SC (Cyclic Prefix Single Carrier) transmission system is used as reference [5].

B. Overlapping Data Estimation

As mentioned before, the use of guard periods reduces the maximum data throughput of the transmission system as no user data can be transmitted while transmitting guard periods. Furthermore, the guard periods increase the transmission energy that is required for one data block.
But what happens if we omit the guard periods and still perform block-wise data estimation in the receiver? Of course the resulting IBI will corrupt the estimated data, but as the channel length is not infinite, we can expect that the distorting influence of the neighbouring blocks must be more significant in the border parts of the estimated blocks [3],[5]. To verify this, the ensemble-averaged estimation error for three neighbouring blocks is depicted in Figure 1 (dashed line). One can see that the estimation error is significantly smaller in the middle parts of the data blocks. This bathtub like error distribution can be exploited by using overlapping data blocks instead of neighbouring blocks, which means that the data symbols in the border part of block N are again estimated as the middle part of block $N+1$. The estimation error can then be reduced by omitting the more erroneous outer parts of each block and keeping the middle parts for further processing.

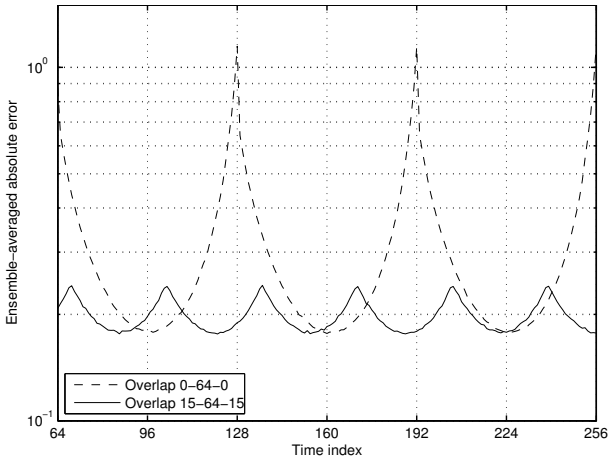

Fig. 1. Error distribution for overlapping data estimation

The resulting ensemble-averaged estimation error is depicted in Figure 1 (solid line).
The notation $(15-64-15)$ gives the size of the pre lap P^e_{pre}, the block size B^e and the post lap P^e_{post}, i.e. for the example the block consists of $B^e = 64$ symbols and the first and last 15 symbols are omitted. The superior indices are used to differentiate the parameters used for overlapping estimation (e) and overlapping decoding (c).
To allow the use of efficient FFT based EVD algorithms for solving the MMSE equation (2), the overlapping block matrices are cyclically extended. This also means that the underlying signal processing structure is similar to known CP-OFDM systems, so the effort for new system designs can be reduced. The resulting overlapping detection scheme is depicted in Figure 2.

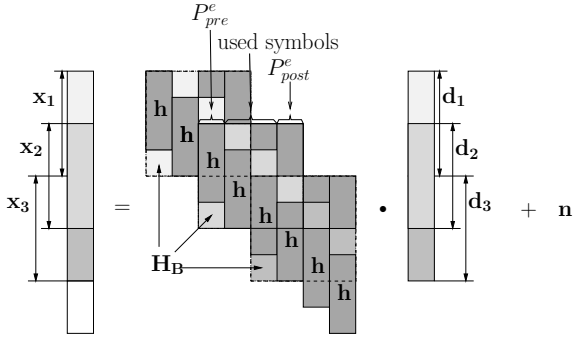

Fig. 2. Data estimation with overlapping submatrices

C. Overlapping Viterbi Decoding

Block signal processing is also known in the context of error correcting channel codes. In case of convolutional channel codes, which are used in many current communication systems, a block structure can be realized by periodically inserting zero sequences into the input data stream. These zero symbols work like guard periods and are called initialization and termination sequences [7].

The application of data blocking in the channel decoder without using initialization/termination sequences of course leads to significant decoding errors. But as these channel codes are based on convolution and the convolution depth, which is given by the number of memory elements of the encoder, is limited, we can expect a similar behaviour as described for the overlapping data estimator. In [4] and [9] it was shown that overlapping Viterbi decoding is feasible. The resulting decoder structure is depicted in Figure 3.

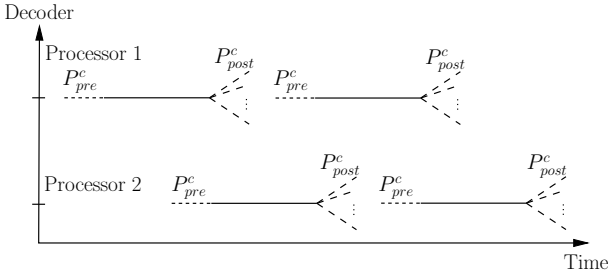

Fig. 3. Channel decoding using overlapping decoders

III. Performance Analysis of the Overlapping Approaches

To determine the performance of the presented overlapping approaches, comparative bit error rate (BER) simulations are performed.

A. Simulation Model

The used simulation model is based on a time variant multi path channel with superposed white Gaussian noise. A detailed description of the channel model can be found in [8].
For the simulations we set the channel length to $L = 17$ taps and presume that the channel is time invariant during the transmission of one data vector. The receiver's channel estimation is considered to be perfect, i.e. the respective impulse response is exactly known in the receiver.
Before the data is passed to the BPSK modulator (Binary Phase Shift Keying) it is encoded and interleaved (interleaving block size $I = 256$ bits). To keep the simulation results clearly arranged, we use a short convolutional code of constraint length $K = 3$ and code rate $R = 1/2$ together with a hard decision Viterbi decoder.

B. Overlapping Data Estimation

The BER performance of the overlapping data estimation scheme for different block lengths and different overlap sizes is shown in Figure 4. The upper bound is given by the case $(0-64-0)$ which means that only neighbouring blocks were estimated, without any overlap. The lower bound is determined by a CP-SC system with a block size of 128 BPSK symbols and a cyclic prefix length of $L-1 = 16$ symbols. For small values E_b/N_0 the overlapping systems perform slightly better compared to the SC-CP system. This is caused by the cyclic prefix, which increases the required transmission energy per data bit. For larger values E_b/N_0 however the remaining IBI

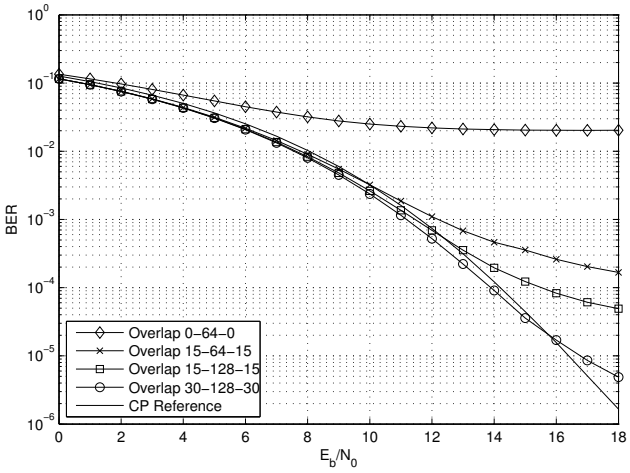

Fig. 4. BER performance for overlapping estimation using different parameter sets

will cause a residual error floor, whose height depends on the block size and overlapping length. On the one hand the BER performance of the system can be improved by increasing B^e and P^e, but on the other hand the computational complexity will also be increased in this case. This means, a trade-off between BER performance and computational complexity has to be found. As these dependencies are still a topic of research, a system with parameters $(15-128-15)$ is exemplarily chosen for further comparisons.

C. Overlapping Viterbi Decoding

In case of overlapping channel decoding we can use properties that are known for the unterminated Viterbi decoder. This means, as we do not use any termination sequences, the overlapping part in the end of the data block corresponds to the truncation length of the convolutional code, which is normally given as $5K$ decoded symbols in literature [7]. A similar relation can be found for the overlapping part in the beginning of the data block, which corresponds to the acquisition length of the convolutional code. It was shown in [4] that using a length of $5K$ for truncation and acquisition length will only affect the performance of the convolutional decoder insignificantly, i.e. we set $P^c_{pre} = P^c_{post} = 5K$.

To verify this, BER simulations for different overlapping lengths are given in Figure 5. If $P^c_{pre} = P^c_{post} > 5K = 15$ no degradation in BER performance is visible for the used code. Accordingly, in contrast to the overlapping detection scheme the proposed decoding scheme does not cause a residual error floor.

D. Combined Overlapping Approaches

For the final block signal processing receiver structure, the overlapping approach has to be applied to both, estimation and channel decoding. As the used channel decoder will not cause any additional BER performance loss (see Section III-C), we can expect that the shape of the BER curve will be mostly influenced by the overlapping detection scheme. The simulated

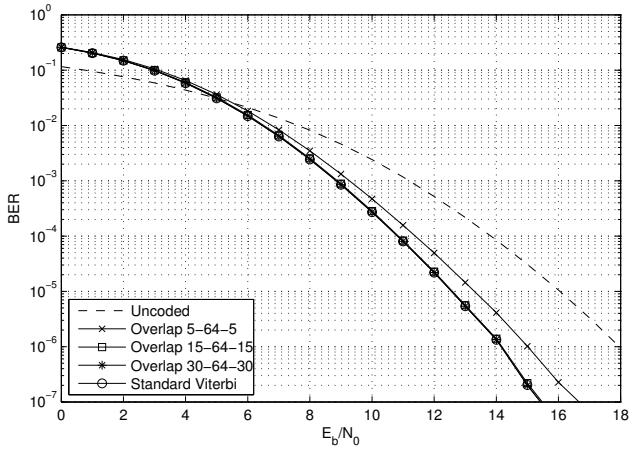

Fig. 5. BER performance for overlapping decoding using different parameter sets

BER performance for the parameters $(15-128-15)$ in case of overlapping estimation and decoding can be found in Figure 6. One can see that the BER degradation caused by the overlapping structure is negligible for $E_b/N_0 < 12$ or $BER > 10^{-5}$. This means that using overlapping techniques for estimation and decoding can increase the data throughput of a wireless transmission system significantly with only marginal drawbacks in BER performance. The computational complexity is slightly increased, as the data in the overlapping part has to be estimated twice, but the impact on processing speed can be countervailed by parallel signal processing, which is based on the block structure. The parallel implementation aspects are addressed in Section IV.

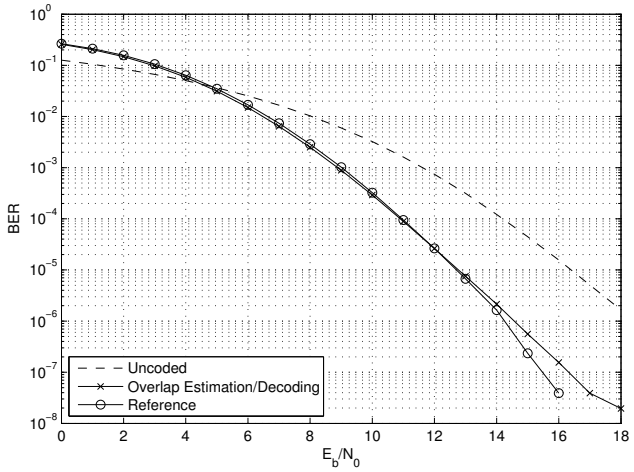

Fig. 6. BER performance for combined overlapping estimation/decoding

IV. Implementation Aspects

The presented block processing structure allows parallel implementations of the required signal processing tasks, i.e. data detection, deinterleaving and channel decoding. The speed-up of the parallel implementation is strongly affected by the amount of communication between the processors [10]. However, due to data dependencies the avoidance of inter processor communication is hard to realize or comes along with a large computational overhead. We will present three different implementation methods, which offer different levels of flexibility regarding the system parameters and communication requirements, while offering identical BER performance.

A. Method A

The most flexible implementation method is based on three independent blocks for data estimation, deinterleaving and channel decoding, as depicted in Figure 7. Estimation and decoding can be parallelized, but communication and synchronization of the processing blocks is required. However there are no restrictions regarding the system parameters, i.e. interleaving block size, overlapping length and estimation/decoding block size can be chosen arbitrarily.

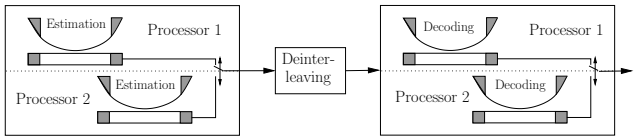

Fig. 7. Implementation for method A

Complexity: As mentioned before, the signal processing structure of the proposed estimation scheme is identical to the one used for CP-SC, except that we use overlapping instead of guard periods to obtain the block processing structure. This means, for a given data vector of length V we have to process $V/(B - P^e_{pre} - P^e_{post})$ blocks instead of V/B blocks for the CP-SC system. The computational overhead factor can then be calculated as follows:

$$M^e_A = \frac{\frac{V}{B^e - P^e_{pre} - P^e_{post}}}{\frac{V}{B^e}} = \frac{B^e}{B^e - P^e_{pre} - P^e_{post}}. \qquad (3)$$

We can use the same approach for calculating the computational overhead factor M^c_A for the overlapping Viterbi decoder. As mentioned in Section III-C we use $P^c_{pre} = P^c_{post} = 5K$ to prevent a loss of decoding performance. In this case the overhead factor is given as

$$M^c_A = \frac{B^c}{B^c - 10K}. \qquad (4)$$

For our example with parameters $(15 - 128 - 15)$ for both estimation and decoding the overhead factors are $M^e_A = M^c_A = 1.30$. The interleaving task is not affected by the new receiver structure, but has to be considered here due to its impact on data dependencies.

B. Method B

Embedding the deinterleaver into the estimation processor leads to method B, which is depicted in Figure 8. By this embedding the communication effort can be reduced and the deinterleaving process can be realized in the parallel structure. However this leads to a constraint regarding the interleaving

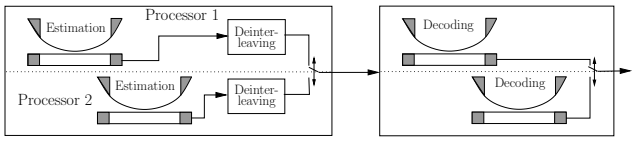

Fig. 8. Implementation for method B

block length and the estimation block length respectively, which are now connected by

$$B^e - P^e_{pre} - P^e_{post} \equiv nI, \tag{5}$$

where n is an arbitrary integer number greater than zero.
Complexity: The complexity of method B is identical to method A, in principle. However, the interleaving length has to be adjusted to the used transmission channel and cannot be reduced arbitrarily. This means, the minimum output block size is determined by the chosen interleaving length.

C. Method C

An entire parallel receiver structure without any inter processor communication requirements can be obtained by implementing decoding and estimation on the same processor. This however leads to complications regarding the overlap of the channel decoder. In method A and B overlapping parts of the estimator output are processed by the parallel Viterbi decoders, which is not possible in this case as we try to prevent inter processor communication. We solve this problem by modifying the data estimator in a way, such that the output block size is identical to the input block size, i.e. the overlapping parts are not omitted after estimation. The corrupted outer parts of the estimator output block are then re-used as overlapping parts for the channel decoder and the extraction of the inner part of the data block is realized after the decoding process. The deinterleaving is realized block-wise between data estimator and channel decoder, i.e. the block size and overlapping lengths have to be integer multiples of the interleaving length. The entire processing structure is depicted in Figure 9. Even if the used overlapping parts for

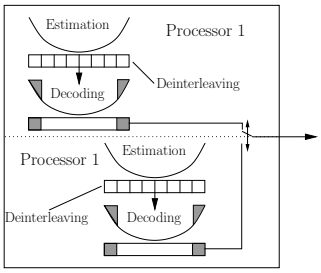

Fig. 9. Implementation for method C

the channel decoder contain the most corrupted symbols from the estimator, the BER performance is not decreased, because the minimum overlapping length is $P = I$, which is longer compared to method A and B.
Complexity: The complexity of method C is determined by the interleaving length I, i.e. both overlap length and block length have to be integer multiples (p, q) of I. As the block length for estimation and decoding have to be identical the overhead factors are also identical and can be calculated as

$$M_C^{e,c} = \frac{B}{B - P_{pre} - P_{post}} = \frac{pI}{pI - 2qI} = \frac{p}{p - 2q}. \tag{6}$$

Based on a reduced interleaving block size of $I = 64$ we choose a block length of $B = 8I = 512$ and an overlapping length of $P = 1I = 64$. The computational overhead factor is then given as $M_C = 1.33$, which is a moderate value and would allow a maximum speed-up of 50% on a two processor system. Depending on the transmission channel, however, the interleaving length might be increased to prevent performance loss of the convolutional code. This will also increase the computational overhead for the presented implementation method and can reduce the speed-up. Hence, the resulting speed-up of the receiver's signal processing structure strongly depends on the chosen system parameters.

V. Conclusions

In this paper a data estimator and a channel decoder, both based on overlapping block processing schemes, were combined in a receiver. The resulting receiver architecture allows parallel block-wise signal processing. As the overhead in terms of additional mathematical operations is moderate, a high speed-up can be achieved by this parallel structure. Three different implementation methods were presented, which lead to different levels of flexibility and inter processor communication efforts.
Furthermore the avoidance of guard periods increases the data throughput of the presented system, as more payload can be transmitted in a given time interval. Both attributes make the presented system especially suitable for high speed wireless communication systems.

References

[1] Z. Wang and G. B. Giannakis: Wireless Multicarrier Communications, *IEEE Signal Processing Magazine*, pp. 29–48, May 2000.
[2] IEEE: Std 802.11a. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. High-Speed Physical Layer in the 5 Ghz Band, 1999.
[3] C. V. Sinn and J. Götze: Avoidance of Guard Periods in Block Transmission Systems, *Proceedings of IEEE Signal Processing Advances in Wireless Communications*, Rome, Italy, June 2003.
[4] G. Fettweis and H. Meyr: High-speed parallel Viterbi decoding: Algorithm and VLSI-architecture, *IEEE Communications Magazine (ComMag)*, 29(5):46 - 55, May 1991.
[5] C. V. Sinn: Computationally efficient block transmission systems with and without guard periods, *Signal Processing*, 87(6):1421-1433, Elsevier, 2007.
[6] R. van Nee and R. Prasad: *OFDM for Wireless Multimedia Communications*, Artech House Publishers, ISBN 0-89006-530-6, 2000.
[7] Shu Lin & Daniel J. Costello, Jr.: *Error Control Coding*, 2. Edition, Pearson Prentice Hall, 2004.
[8] K. Ruttik et. al.: A Wideband Radio Channel Model for Simulation of Chaotic Communication Systems, *Proceedings of ECCTD*, pp. 302-305, Budapest, August 1997.
[9] G. Fettweis: *Parallelisierung des Viterbi Decoders: Algorithmus und VLSI-Architektur*, Band 144 of the VDI Fortschrittsberichte, VDE Verlag, Düsseldorf, 1990.
[10] V. Kumar, A. Grama, A. Gupta and G. Karypis: *Introduction to Parallel Computing, Design and Analysis of Algorithms*, Benjamin/Cummings, 1994.